

1. A SZÁMÍTÓGÉPES GRAFIKA ALAPFOGALMAI ÉS FELHASZNÁLÁSI TERÜLETEI

1. Hogyan határozzák meg a számítógépes grafika fogalmát?

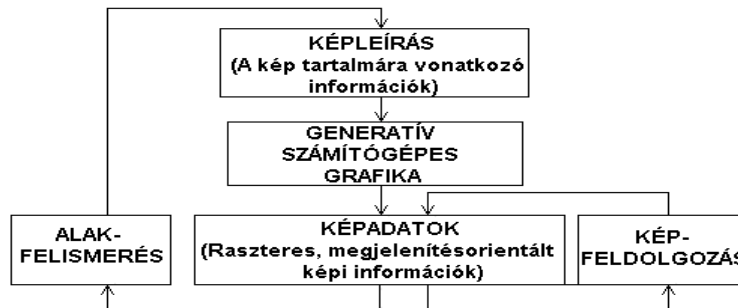
Legáltalánosabban a számítógépes grafika (computer graphics) alatt a számítógép és a grafikus perifériák között megvalósított adatkonvertálási módszereket és eljárásokat értjük.

Ez alapján három, relatíve önálló részterületet különböztethetünk meg:

- a generatív számítógépes grafikát (generative computer graphics),
- a képi információk számítógépes feldolgozását (image processing),
- a számítógépes képelemzést, alakfelismerést (picture analysis).

2. Mit nevezünk generatív számítógépes grafikának?

Ezek viszonyát mutatja az alábbi ábra.



A generatív számítógépes grafika, az alakfelismerés és a képfeldolgozás összefüggései

A generatív számítógépes grafika ezek szerint a képi információ tartalmára vonatkozó képleírási adatok (és nem a kép) alapján, algoritmusokkal állít elő pl. a monitor képernyőjén megjeleníthető képeket.

3. Mi a számítógépes grafika tárgya?

A generatív számítógépes grafikában a képleírások valamely szabályrendszer (szintaxis) szerint határozzák meg a grafikus objektumokat.

Például a POV-Ray programcsomag segítségével egy formális nyelven a modelltérben (3D lebegőpontos világkoordinátarendszer) 3D objektumokat definiálunk, majd róluk fotorealisztikus képeket készítünk.

A számítógépes grafika tárgyát – az előbbiekre is figyelemmel – a következőképpen határozhatjuk meg: a számítógépes grafika alatt a két- (2D) és három- (3D) dimenziós grafikus objektumok számítógépes generálását, tárolását, feldolgozását és megjelenítését értjük.

4. Mit jelent az alakfelismerés?

A számítógépes alakfelismeréssel a raszteres képeken lévő grafikus objektumok azonosítását végezzük el. Ezekkel – a legtöbb esetben bonyolult matematikai eljárásokkal – a raszteres képből kinyerjük a képleíráshoz szükséges lényegi információkat.

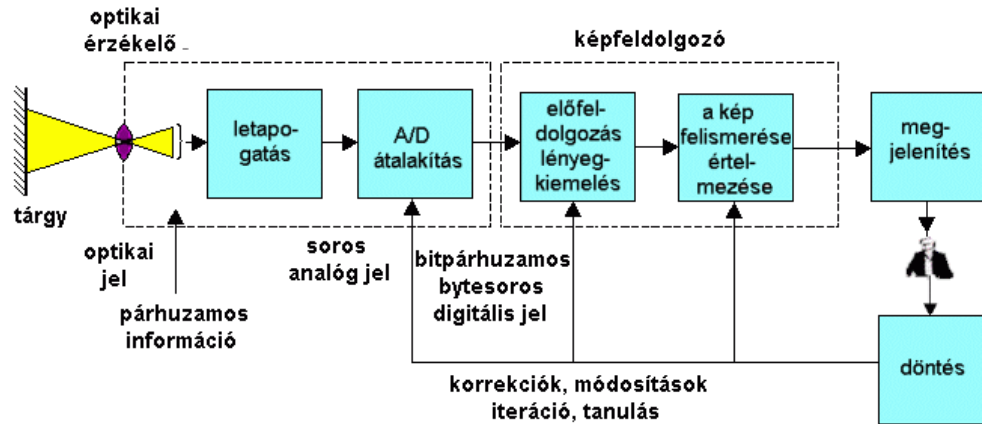
Az alakfelismerés és a képfeldolgozás módszereinek és eljárásainak együttesét nevezzük digitális képfeldolgozásnak (image processing).

5. Mit nevezünk képfeldolgozásnak?

A képfeldolgozás mindazon számítógépes eljárások és módszerek összességét jelenti, amelyekkel a számítógépen tárolt képek minőségét valamilyen szempont szerint javítani lehet (pl. élek kiemelése), és ezáltal továbbfeldolgozásra alkalmasabbá válnak.

A képfeldolgozó rendszerek a képeket nem generálják, hanem inputként kapják meg (pl.

digitalizált fotók, mesterséges holdfelvételek stb. formájában). A képfeldolgozás alapegysége a raszteres kép, amely $n \times m$ képpontból (pixelből) áll. A képpontokat szürkeség vagy színinformációkkal jellemezzük, ezek együttese alkotja a képadatokat.

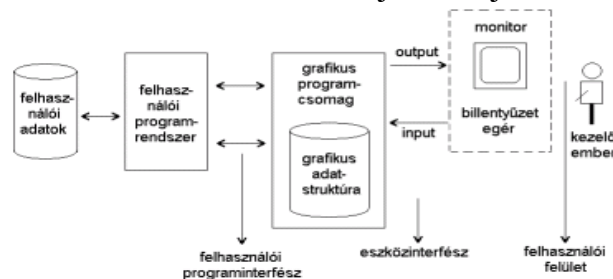


6. Mit jelent a grafikus objektumok modellezése?

A számítógépes grafikában és a képfeldolgozás során nem a valódi objektumokat, hanem azok egy modelljét dolgozzuk fel. A modellalkotás során megpróbáljuk a grafikus objektum lényegi jellemzőit megragadni, és az így absztrakcióval képzett számítógépes modellt algoritmusokkal dolgozzuk fel.

7. Sorolja fel milyen elemekből épül fel egy grafikus rendszer!

A grafikus rendszerek ennek megfelelően kialakult architektúráját mutatja az alábbi ábra.



Látható, hogy egy grafikus rendszer három fő részből,

- az alkalmazás-specifikus felhasználói programrendszerből (CAD-rendszer, térképészeti rendszer stb.),
 - a szabványosított grafikus programcsomagból (pl. DirectX) és
 - a grafikus hardverből (pl. grafikus kártya és monitor)
- tevédik össze.

8. Mi a szerepe a szabványos csatolóknak (interface) a grafikus rendszerekben?

Az egyes rendszerrészeket szabványos interfészek vagy csatolók kapcsolják össze, ami által az egyes rendszeregységeket (pl. újabb verziójú szoftver vagy fejlettebb grafikus periféria megjelenése esetén) úgy tudjuk lecserélni, hogy a többi rendszerkomponenst nem kell megváltoztatni. Ezek a csatolók a következők:

9. Mit jelent az API, és mi a feladata a grafikus rendszerekben?

A grafikus programcsomag és a felhasználói programrendszer között helyezkedik el a felhasználói programcsatoló vagy API (Application Program Interface). Ez általában azt is biztosítja, hogy a grafikus programcsomaggal különböző programnyelven tarthassanak kapcsolatot a felhasználói programok (language binding).

10. Mi a feladata a hardver csatolónak (device interface), és milyen rendszerelemek tartoznak e csatolóhoz?

A grafikus programcsomag egy-egy konkrét grafikus perifériával a hardvercsatolón (device interface) keresztül tartja a kapcsolatot.

Ide tartoznak a különféle eszközmeghajtók (device driver) és a BIOS megfelelő része (video ROM-BIOS);

11. Mit jelent a GUI, és mikor vált szabvánnyá?

A grafikus rendszerrel a felhasználó egy szabványos grafikus felületen tarthat kapcsolatot. Ez a GUI (Graphical User Interface), mely az 1980-as évek második felében vált szabvánnyá (X11 és X-Windows 1987).

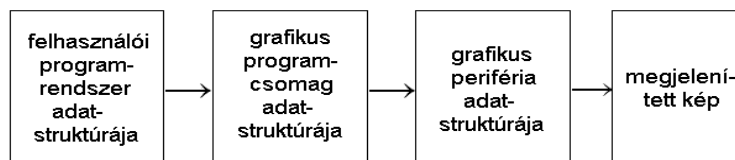
12. Mit nevezünk primitívnek?

A grafikus programcsomag a képet saját rendszerében kezelt grafikus objektumokból állítja elő. A legkisebb, a grafikus programcsomag által már további részekre nem bontható elemi grafikus objektumokat primitíveknek nevezzük. A grafikus programcsomag ezekből a primitívekből állítja elő a komplexebb objektumokat. Példák primitívekre:

3D vektorgrafikában egy térbeli egyenesszakasz, rasztergrafikában egy kör.

13. Milyen transzformációk szükségesek a modelladatok képernyőn történő megjelenítéséhez?

A képernyőn történő megjelenítéshez a felhasználói programrendszer modelladatait először a grafikus programcsomag primitívjeiből felépített objektumokká, majd raszteres képpé kell transzformálni (lásd az ábrát).



A modelladatok megjelenítéséhez szükséges transzformációk

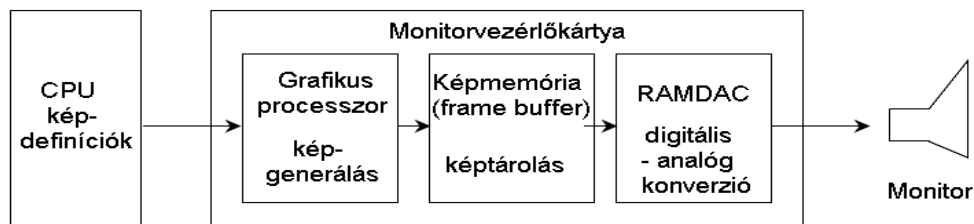
14. Mit nevezünk pixelnek?

A pixel (picture element) magyarul a kép elemi, ezért tovább fel nem bontható részét jelenti.

A raszteres képernyőn történő képmegjelenítés feladatainak egyre jelentősebb része napjainkban már a grafikus kártyára hárul, ezáltal tehermentesítve a processzort.

15. Mi a monitorvezérlő kártya részegységeinek feladata raszteres képek megjelenítése során?

Az együttműködő hardveregységek közötti, a raszteres kép megjelenítése során létrejövő feladatmegosztást mutatja az ábra.



A hardver egységek közötti feladatmegosztás a raszteres kép megjelenítésében

A processzor (CPU) által átadott képdefiníció alapján a képpontok kiszámítása raszteres primitívekkel történik meg, és ennek eredményét a grafikus processzor a képmemóriába (frame buffer) írja be. A képmemória két porttal rendelkezik, ezt a grafikus processzor és a digitális-analóg konverter konkurens módon használja.

A RAMDAC digitális-analóg konverter a képmemóriát olvasva vezérli a monitor kirajzoló elektronágyúit és így jelenik meg a kép a monitor képernyőjén.

16. Hogy állítja elő a színes monitor a pixeleket?

A pixelek tulajdonképpen a képernyőn három darab, az alapszíneknek (piros, zöld, kék) megfelelő részpontból állnak, amelyet a CRT (Cathode Ray Tube) monitor esetében három elektronsugár „gyújt fel” a monitor foszforrétegén

17. Jellemezze a színtereket, mint vektortereket!

A színek egy háromdimenziós matematikai struktúrát alkotnak, azaz megfeleltethetők egy 3D-s vektortér vektorainak. Ezt a vektorteret színtérnek hívjuk, a tér egyes vektorait színvegyértéknek. A színkeverésnek két alapvető módja van, aszerint, hogy elsődleges vagy másodlagos fényforrást modellezünk.

Az összeadó, vagy additív színkeverésnél a vörös, zöld és kék alapszínekből vett meghatározott mennyiségeket adunk össze és így kapjuk a különböző színárnyalatokat. Ezzel az úgynevezett elsődleges fényforrások színeit tudjuk előállítani.

A kivonó vagy szubsztraktív színkeverésnél az alapszínek komplementereiből (ciánkék, bíborvörös, sárga) állítjuk elő a színeket. Ezzel lehet modellezni a különböző tárgyak által visszavert fényt. (Kivonó színkeverés például színszűrőkkel hozható létre. Ha egy lámpa elé piros, illetve sárga színszűrőt helyezünk, piros, illetve sárga fényt látunk. Ha mindkét színszűrőt feltesszük, a keverékszín narancs lesz, a fényintenzitás csökken.)

18. Mit nevezünk additív színkeverésnek?

A színkeverésnek két alapvető módja van, aszerint, hogy elsődleges vagy másodlagos fényforrást modellezünk.

Az összeadó, vagy additív színkeverésnél a vörös, zöld és kék alapszínekből vett meghatározott mennyiségeket adunk össze és így kapjuk a különböző színárnyalatokat. Ezzel az úgynevezett elsődleges fényforrások színeit tudjuk előállítani.

19. Mit nevezünk szubsztraktív színkeverésnek?

A kivonó vagy szubsztraktív színkeverésnél az alapszínek komplementereiből (ciánkék, bíborvörös, sárga) állítjuk elő a színeket. Ezzel lehet modellezni a különböző tárgyak által visszavert fényt. (Kivonó színkeverés például színszűrőkkel hozható létre. Ha egy lámpa elé piros, illetve sárga színszűrőt helyezünk, piros, illetve sárga fényt látunk. Ha mindkét színszűrőt feltesszük, a keverékszín narancs lesz, a fényintenzitás csökken.)

20. Mi az RGB színtér?

Az additív illetve szubsztraktív színkeverésnek megfelelően képezhetjük a számítástechnikában legelterjedtebben használt RGB, CMY és CMYK színtereket:

Az RGB színtér a vörös, a zöld, a kék (Red, Green, Blue) alapszínekből kikeverhető színeket tartalmazza, az additív színkeverés modellezéséhez használjuk.

21. Mi az CMY színtér?

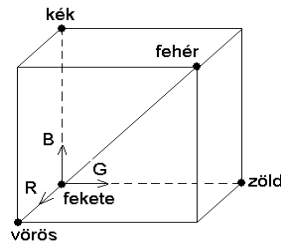
A CMY színtér a ciánkék, a bíborvörös, a sárga (Cyan, Magenta, Yellow) alapszínekből kikeverhető színeket tartalmazza, a szubsztraktív színkeverés modellezéséhez használjuk.

22. Mi jellemzi a CMYK színteret?

A CMYK színtér megegyezik a CMY színtérrel, azzal a különbséggel, hogy a CMY színtér alapszíneihez még hozzáadjuk a „tisztá” fekete színt is. Ennek az az oka, hogy a CMY alapszínek keverésével csak sötétszürke színt tudunk előállítani, és a nyomdatechnikában a teljesen fekete színre is szükségünk van.

23. Adja meg az RGB és CMY színterek közötti átszámítás szabályait!

Az RGB és CMY színtereket egy egységkockával szokták ábrázolni (lásd ábrát).



Az RGB színtér
szemléltetése
egységkockával

Az RGB és CMY színterek alapszíneinek arányai közötti átszámítás kölcsönösen egyértelmű:

$$[c, m, y] = [1, 1, 1] - [r, g, b]$$

$$[r, g, b] = [1, 1, 1] - [c, m, y]$$

Az RGB és CMYK színterek színei között viszont nem lehetséges kölcsönösen egyértelmű megfeleltetés.

24. Mi jellemzi a HSB (HSV) színteret?

Az RGB, a CMY és a CMYK színterek felépítését alapvetően technikai szempontok határozták meg. Ezért olyan színtereket is kialakítottak, amelyek jobban alkalmazkodnak az emberi érzékeléshez.

Egy ilyen a HSB színtér (nevét a Hue = színárnyalat, Saturation = színtelítettség, Brightness = világosság angol kifejezések rövidítéséből kapta), amelyben az RGB alapszínek mellett a színek előállításához a színtelítettség és a megvilágítás erősség értékeit is felhasználhatjuk. A színárnyalat, azaz maga a szín, a szemünkbe jutó fény domináns hullámhosszától függ. A színtelítettség az érzékelt fényben megtalálható fehér fény százalékos összetevőjétől függ. A világosságot vagy fényerősséget a szemünkbe érkező fényenergia határozza meg. Megjegyezzük, hogy a HSB színtért esetenként HSV-nek (Hue Saturation Value) is szokták nevezni.

A HSB színtért egy hengerkoordináta-rendszerben ábrázolhatjuk az ábra szerint. A HSB színtérnél a kör 360°-jából egy konkrét szögértékkel jellemezhetjük az RGB színek közötti átmeneteknek megfelelő színárnyalatokat. A kör középpontjától mért távolsággal fejezhetjük ki a színtelítettséget, és a henger alsó alapkörétől mért távolság adja meg a fényerősséget.

25. Mi a True Color színkódolás?

Ha három alapszínünk van – mint például az RGB színtérben –, akkor csatornánként meg kell határozni, hogy hány biten kódoljuk az adott alapszín intenzitásértékét.

Az úgynevezett High Color színmegjelenítés esetén a három RGB csatornát összesen 16-biten kódoljuk. Ez összesen 2^{16} , azaz közelítőleg 32000 különböző színárnyalat megjelenítését teszi lehetővé! Napjainkban legelterjedtebb a True Color (igazi színek) háromcsatornás színkódolás, amikor a három RGB alapszín intenzitását $3 \times 8 = 24$ biten kódolják, ami 2^{24} , azaz kb. 16 millió színárnyalat megjelenítését teszi lehetővé. Az egyes alapszínek intenzitását a felhasználó ebben az esetben egy 0 és 255 közötti számérték beállításával adhatja meg.

26. Definiálja a rasztergrafikus rendszert!

A grafikus képek számítógépes feldolgozása a kezdetekben csak a különböző rajzoló programok alkalmazását jelentette. Ezek rasztergrafikus alapon működtek, azaz képpontokból (pixelekből) álló képek feldolgozását végezték.

A számítógépes grafikában azokat a raszteres, azaz képpontokból (pixelekből) álló képet generáló és feldolgozó rendszereket, amelyeknél a képi információ csak képenként kereshető vissza, és a kép tartalma csak a teljes kép felülírásával módosítható, rasztergrafikus rendszereknek nevezzük.

A számítógépes grafikában azokat a rendszereket, amelyek a grafikus objektumokat egy lebegőpontos világkoordináta-rendszerben modellezik, vektorgrafikus rendszereknek nevezzük.

A rasztergrafika és vektorgrafika közötti legfontosabb különbséget a grafikus objektumok képenkénti, illetve adatbázisszerű kezelése jelenti (lásd az ábrát).

RASZTERGRAFIKA	VEKTORGRAFIKA
a pixelekből álló képet tároljuk, csak a kép kereshető vissza, a képen látható objektumok nem	az objektumok leírását (modelljét) önállóan tároljuk, ezek egyedileg is visszakereshetők

27. Mit nevezünk vektorgrafikus rendszernek?

ua. mint az előző

28. Melyek a vektorgrafika és rasztergrafika közötti legfontosabb különbségek?

A vektorgrafika gyors fejlődését jórészt a rasztergrafika és a vektorgrafika különbségei magyarázzák:

A vektorgrafika egy 3D-s (korábbi változatai 2D-s) lebegőpontos világkoordináta-rendszert használ, ezáltal lehetővé teszi a geometriai pontosságú szerkesztést és transzformációkat. Így sokkal alkalmasabb a rasztergrafikánál a mérnöki és tudományos munka támogatására.

A vektorgrafika absztrakt modelltérbeli tárgyakkal dolgozik. Ezek önálló objektumok (entitások), amelyekkel műveleteket lehet végezni a képernyőn való megjelenítéstől függetlenül is. A felhasználói felület, a grafikus programcsomagok és a megjelenítő hardver szabványosítása ugyanakkor lényegében bármely elterjedt számítógép-konfiguráción lehetővé teszi a modelltérbeli vektorgrafikus objektumok raszteres képen való megjelenítését akár több nézőpontból (kameraállásból) is. Ezzel szemben egy rasztergrafikus képet – lényegét tekintve – csak a kép felülírásával tudjuk módosítani.

A vektorgrafikában a grafikus objektumokat adatbázisban tárolják, amely lehetővé teszi az egyes testek, tárgyak modelljeinek egyedi visszakeresését és az ezek közötti kapcsolatok rögzítését és kimutathatóságát. A rasztergrafikában viszont nincs lehetőség az egyes képeken belüli grafikus rajzelemek önálló visszakeresésére és kezelésére.

29. Határozza meg a virtuális valóság fogalmát!

Virtuális valóság alatt elképzelt, vagy méretük, illetve távolságuk miatt láthatatlan, veszélyességük miatt megközelíthetetlen világok valósághű, interaktív modellezését és megjelenítését értjük számítógépen.

Ezeket a mesterséges, háromdimenziós hatást keltő világokat a rendszert használó ember bejárhatja, felfedezheti.

30. Mi különbözteti meg a grafikában a játékfejlesztést a filmkészítéstől?

A játék a filmmel ellentétben interaktív. Mindössze ez az interaktivitás ill. a real-time válaszidő az a különbség, ami a grafikában a játékfejlesztést és a filmkészítést jelenleg megkülönbözteti. A film és az interaktív játék házasságára a jövőben a technológiai fejlődés minden bizonnyal lehetőséget fog biztosítani (fotorealisztikus virtuális valóság 3D sztereo hanggal, interaktív film), de ez számos etikai, társadalomtudományi, művészeti kérdést vethet fel, amely már régóta foglalkoztatja a tudósokat és a művészeket.

Példa erre a Harry Potter című játékprogram

31. Mit értünk fotorealisztikus ábrázolás alatt a számítógépes grafikában?

A számítógépes grafika fejlődésében fontos állomást jelentett a képek fotorealisztikus ábrázolása, amely a felhasználói területeket jelentősen kiszélesítette.

Fotorealisztikus képábrázolásról akkor beszélünk, ha a számítógépes grafikával generált képeket gyakorlatilag nem lehet megkülönböztetni a fénykép vagy videofelvételtől. Ennek egyik következménye az is, hogy a számítógépes grafika és multimédia közötti határ elmosódik.

32. Mutasson be néhány jellemző tendenciát, mely az Internet és a számítógépes grafika kapcsolatára jellemző!

A világhálózat növekvő szerepére a számítógépes grafika fejlődésében további példákat is találhatunk:

- A jelentősebb grafikus programcsomagokkal ma már kivétel nélkül lehetséges a hálózati csoportmunka, azaz a fizikailag elkülönült helyen dolgozó szakemberek például közös építészeti terveket készíthetnek.
- Az Internet hozzájárul a grafikus adatbázisokban lévő szellemi értékek felhalmozódásához. Példák erre a clipart könyvtárak, a grafikus adatbázis elemeket tartalmazó szimbólumkönyvtárak, fontkönyvtárak (lásd a következő lapon).
- Terjed a Computer Based Training a hálózaton, azaz egyre több a grafikus, multimédiás interaktív oktatóprogram az Interneten.
- A különböző programcsomagok között a képfájl szabványok általános elterjedésével az adatcsere is egyre inkább lehetővé válik a világhálón.
- A fontosabb grafikus programcsomagok lényegében azonos kezelő felületet kínálnak a felhasználó számára, ezért egy programcsomag szolgáltatásai könnyebben megtanulhatóak.

33. Milyen műszaki követelményeknek kell megfelelni a világhálózatnak a számítógépes képfeldolgozás tömeges elterjedéséhez?

Természetesen ahhoz, hogy ez a fejlődés kiteljesedjen, meghatározott műszaki követelményeknek is meg kell felelnie a hálózatnak. Ilyen például:

- a nagy sávszélességű adatátvitelre való képesség;
- olyan médiaszerverek elterjedése, amelyek több száz videostream egyidejű kiszolgálására is képesek;
- szabványos, multimédiás és grafikus üzemmódra is alkalmas kliens hardverek és operációs rendszerek használatának általánossá válása.

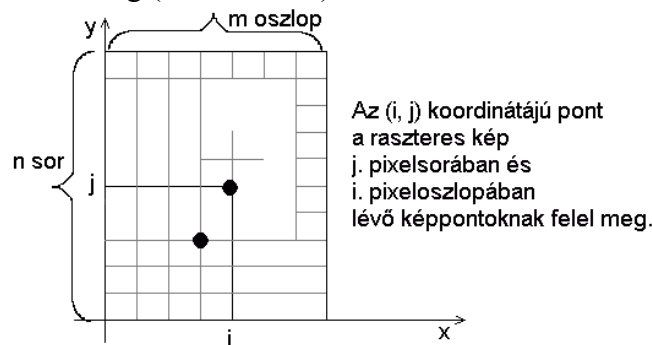
2. RASZTERGRAFIKA

34. Miből épül fel a raszteres kép?

A képpontokból (pixelekből) felépülő képet raszteres képnek nevezzük, ezek számítógépes feldolgozását pedig rasztergrafikának. Megjegyezzük, hogy a „képpont” megfogalmazás esetenként félrevezető lehet, mivel – helytelenül – a geometriai értelemben kiterjedés nélküli pontra is gondolhatunk.

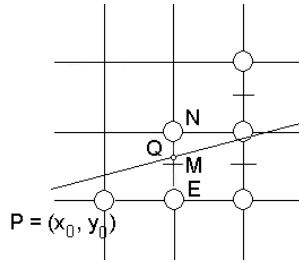
35. Jellemezze a rasztergrafika modellterét!

A rasztergrafika modelltere egy kétdimenziós egész koordináta-rendszer, melyben a képpontoknak egészértékű koordinátapontok felelnek meg (lásd az ábrát).



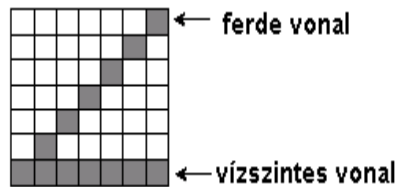
36. Mi a Bresenham-algoritmus lényege?

Az egyik legismertebb algoritmus a Bresenham-féle középpontos vonalalgorithmus. Ezt az eljárást az egyenes példáján mutatjuk be, de alkalmazható görbe vonalak rajzolására is. Ennek lényege, hogy a raszteres képen „oszlopirányban” haladva minden egész értékű „x”-koordinátában a matematikai egyeneshez függőlegesen legközelebbi pontot választjuk.



37. Milyen fényerőproblémák jelentkeznek ferde vonalak képernyőn való megjelenítésekor, és mi az anti-aliasing?

Egy másik problémával találjuk magunkat szembe, ha nemcsak vízszintes és függőleges vonalakat húzunk a képernyőn. Nézzük meg például egy ferde egyenes erősen kinagyított képét, amikor az úgynevezett lépcső effektus fellép (lásd az ábrát). Ezek a ferde vonalak már relatíve nem túl nagy felbontásnál is észrevehetők.



A vonalak „lépcsőzetes” képe raszteres képernyőn

A raszteres egyenesszakaszok ábrázolásának további gondja, hogy a ferde és vízszintes vonalak fényereje is eltér egymástól.

Ennek oka például az ábra példáján bemutatva, hogy a vízszintes vonal 7 egységnyi hosszú és 7 pixelt tartalmaz, a ferde vonal $7 \cdot \sqrt{2} \approx 10$ egységnyi hosszú és szintén 7 pixelt tartalmaz.

38. Mi a super-sampling eljárás?

Az élsimításnak egy másik, nagyon gyakran használt eljárása az úgynevezett super-sampling. Ennek lényege, hogy például az éleken elhelyezkedő pixeleket felbontják $4 \times 4 = 16$ db további részre, amelyet subpixelnek nevezünk. (Ezek természetesen csak az algoritmusban léteznek, nem valódi képpontok.) Ez lényegében azt jelenti, hogy a raszteres képpontokat elvileg egy nagyobb felbontásnak megfelelően számítjuk ki. Egy megjelenített pixel színe vagy szürkeségértéke ezt követően a hozzá tartozó részpixelkekhez rendelt értékek átlagolásával kerül kiszámításra.

39. Mit nevezünk rasztergrafikus primitívnek?

A rasztergrafikus primitívek olyan, a programcsomagokba beépített rajzelemgenerátorok, amelyekkel a felhasználó tipikus rasztergrafikus objektumokat hozhat létre. Ilyenek például:

- vonalak,
- sokszögek (háromszög, négyzet, téglalap stb.),
- kör és ellipszis,
- szövegek (betűcsalád, betűtípus, például: vastag, dőlt, keskeny, széles, árnyékolt stb.)

40. Milyen tulajdonságokat lehet hozzárendelni a rajzelemekhez?

A rajzelemekhez tulajdonságokat rendelhetünk hozzá. Ezek lehetnek:

- vonalstílus,
- vonalvastagság,
- szín- és
- terület-meghatározó primitívek esetében kitöltő szín, illetve mintázat.

41. Mi a feladata a kiadványszerkesztő programcsomagoknak?

A kiadványszerkesztő programcsomagok célja, hogy a felhasználó számára a szöveg szerkesztése mellett olyan funkciókat is biztosítson, amelyekkel a kiadványok nyomdai előállításához szükséges előkészítő munka teljes körűen elvégezhető. Ilyenek például: a nyomdai minőségű szedés, tördelés, több hasáb (kolumna) kezelése, fotók, grafikák négy színre bontással történő nyomtatása, a szövegelemek helyének tipográfiai pontosságú meghatározása.

42. Mit nevezünk DTP-nek?

Az előbbiekre figyelemmel a nyomdai kiadványszerkesztést (DTP = Desk Top Publishing) úgy definiálhatjuk, mint nyomdai anyagok előállítását számítógéppel és a megfelelő szoftverrel.

43. Hogyan határozhatjuk meg rasztergrafikus rendszerekben a szövegek formáját?

A szövegek grafikus megjelenítéséhez első lépés a megfelelő betűkészlet megválasztása. A formák meghatározásához a rasztergrafikus rendszerek a következő lehetőségeket biztosítják:

- betűcsalád (Times, Helvetica stb.) meghatározása,
- betűtípus (vastag, dőlt stb.) kiválasztása,
- betűnagyság rögzítése.

44. Mit nevezünk betűcsaládnak?

Betűcsaládnak nevezzük az azonos grafikus jellemzőkkel és formai sajátosságokkal rendelkező betűk összességét. Ha a betűcsaládot kiválasztjuk, akkor egy betűformátumnak megfelelő fontot és egyúttal az ezt generáló programcsomagrészt is rögzítettük.

45. Hogyan kezeli a fontokat a Windows operációs rendszer?

A betűcsaládok – generálásuk elve szerint – vagy vektorosak, vagy raszteresek. A Windows operációs rendszerben például megtalálhatjuk a

TrueType, méretezhető vektorbetűket (kiterjesztésük .ttf) és a bittérképes rendszerfontokat (kiterjesztésük .fon)

46. Miben különbözhetnek egy családon belül a betűtípusok?

A betűcsaládokon belül betűtípusokat különböztethetünk meg. A betűtípusok megtartják a család általános grafikai jellemzőit, de néhány tulajdonságukban eltérhetnek egymástól. Ezek:

- a betűkép sötétebb vagy világosabb megjelenése,
- a betűkhöz tartozó vonalak vastagsága (vékony, normál, félkövér, kövér),
- egyenesállású vagy döntött a betű képe.

47. Milyen műveleteket hajthatunk végre a rasztergrafikus felhasználói felületen?

A rasztergrafikában a kép egy adott részét át lehet helyezni és ekkor az alakzat helyén a háttérszín marad vissza (lásd az alábbi ábrát).

A nagyításnak csak egész értékeket lehet megadni és az erősen kinagyított kép eldurvul (lásd az ábrát).

48. Mit nevezünk clipartnak, és ezek mire használhatók?

Clipartoknak nevezzük azokat a kisméretű képeket, melyeket általában szimbólumként, emblémaként, logóként vagy egyszerűen csak díszítőelemként alkalmazunk egy képen. Ezek formai megjelenésüket tekintve lehetnek sematikus rajzok, rastergrafikával generált kis képek vagy beszkenelt (esetleg átalakított) képek.

49. Mit nevezünk sztereogrammnak?

A sztereogrammok olyan 2D-s képek, amelyek a nézőben olyan érzést keltenek, mintha 3D-s képet nézne.

50. Mit nevezünk morphing-nek, és mi a morphing algoritmusok lényege?

Morphing alatt a számítógépes grafikában azt a transzformációt értjük, amelynek során egy kép alakját folytonosan változtatva „átfolyik” egy másikba.

51. Hogyan kereshetjük vissza és módosíthatjuk a raszteres képeket?

A raszteres képfájlok a számítógép tárolóeszközein általában a következő részekből épülnek fel:

- fejléc – megadja a kép formátumát, méretét pixelekben (esetleg a színpalettát),
- adatrész – pixelenként tartalmazza a színek kódokat.

Ebből azonnal következik a rastergrafika egyik legfontosabb tulajdonsága: a raszteres kép csak teljes egészében kereshető vissza és csak felülírással módosítható. Ez azt jelenti, hogy a raszteres képen lévő elkülönült grafikus objektumokat egyedileg nem tudjuk visszakeresni. Ha egy objektum egy részletét módosítjuk, akkor a teljes képet meg kell változtatni.

52. Milyen fajta raszteres képtípusok vannak?

- bittérképes képek (bitmapped image),
- szürkeárnyalatú képek (grayscale image),
- színpalettával indexelt képek (indexed color image),
- valódi színezetű képek (true color image).

53. Mit nevezünk színpalettának?

A színpalettával indexelt képek pixeljeihez egy színindex értéket rendelünk hozzá, amely egy 256 elemű színtáblázatra hivatkozik, amelyet palettának nevezünk. Ezt a rastergrafikus rendszerek általában a képernyőn is megjelenítik és így biztosítják a felhasználó számára a szín interaktív (például egérrel való rámutatás) kiválasztását. A színpaletta minden 8 bites indexe tehát egy konkrét színárnyalatot határoz meg egy pixel számára.

54. Mit jelent a TrueColor, és hány színárnyalatot lehet kódolni ezzel a módszerrel?

A valódi színezetű (True Color) képek esetében a színtér alapszíneinek megfelelő színcsatornánként adjuk meg az alapszínek intenzitását (True Color raszteres kép). Ez RGB vagy CMY színtér esetén $3 \times 8 = 24$ bit, a CMYK színtér esetén pedig $4 \times 8 = 32$ bit megadását jelenti. Így például RGB alapszín intenzitások keverésével 2^{24} , azaz több mint 16 millió (16 777 216) különböző színárnyalatot tudunk megkülönböztetni.

55. Mikor magas a redundanciája egy közleménynek?

A számítógépes képek tömörítését a redundanciák teszik lehetővé és szükségessé. Kissé leegyszerűsítve úgy értelmezhetjük a redundancia fogalmát, hogy egy közleménynek akkor magas a redundanciája, ha a „megértéséhez” minimálisan szükséges információt kifejező jeleken túlmenően sok felesleges jelet is tartalmaz.

56. Mondjon példákat a kódolási, képi és pszichovizuális redundanciára?

- Kódolási redundancia lép fel akkor, amikor például a fekete-fehér képpontok ábrázolásához 8 bitet, azaz 1 byte-ot használunk fel.
- Képi redundanciát jelent például: a mozgóképeknél és az animációknál jelentős redundancia léphet fel akkor, ha az egymást követő képek csak nagyon kis mértékben különböznek egymástól, és ahelyett, hogy csak az eltéréseket tárolnánk, minden képet teljes körűen kódolunk és tárolunk.
- Pszichovizuális redundanciát jelent például, ha a képminőségbeli eltéréseket (például sok színárnyalattal nagy felbontás) az ember már nem tudja megkülönböztetni.

57. Mi alapján lehetséges a veszteségmentes tömörítés?

A veszteségmentes képtömörítés a kép összes információját megőrzi. Ez úgy lehetséges, hogy ezek az eljárások csak a kódolási és a képi redundanciát szüntetik meg.

58. Mi a GIF?

A raszteres képeknek az Interneten általánosan elterjedt veszteségmentes képtömörítő eljárása a GIF (Graphics Interchange Format). Ez kvázi szabvánnyá a Compuserve hálózat által vált. A GIF tömörítésű fájlok kiterjesztése .gif.

59. Mi teszi lehetővé a veszteséges tömörítést?

A veszteséges tömörítés a képek pszichovizuális redundanciáját használja ki. A tömörítés mértéke aszerint változhat, hogy a képminőségnek milyen követelményeket kell kielégítenie.

60. Mi a JPEG és milyen tömörítés érhető el felhasználásával?

A veszteséges tömörítés legfontosabb algoritmus a DFT, azaz a Diszkrét Fourier Transzformáció. (Ezt szokták DCT-nek, azaz diszkrét koszinusz transzformációnak is nevezni.) Ezt az algoritmust alkalmazza az Interneten legelterjedtebb JPEG nevű veszteséges tömörítés, amely a Joint Photographic Experts Group rövidítésből származik. Az eljárás ISO és CCITT szabvány is. A JPEG tömörítésű fájlok kiterjesztése .jpg.

61. Jellemezze a fraktáltömörítést!

A veszteséges tömörítés egy másik típusa a fraktálok alkalmazásán alapul. A fraktáltömörítés alap gondolata, hogy a természetben előforduló képek általában önhasonló mintákat tartalmaznak.

- A tömörítésnek jelentős a számításgénye. A tömörített kép kifejtése ennél lényegesen gyorsabb.
- A fraktáltömörítéssel lényegesen nagyobb tömörítést lehet elérni, mint pl. a JPEG-gel (1:100)
- A fraktáltömörítésnek jobb a kontúr és színárnyalat visszaadása a korábban ismert tömörítő algoritmusokhoz képest.

A fraktáltömörítésű fájlokat a .fif kiterjesztésről ismerhetjük fel.

62. Melyek a legfontosabb raszteres képfájlformátumok?

BMP = Windows bitmap, független a grafikus kártyától és annak kezelőprogramjától, 24 bites színmélység kezelését biztosítja.

PCX = A Paintbrush festőprogram fájlformátuma eredetileg, jelenleg az egyik legelterjedtebb formátum.

Az RGB színteret a 24 bites színmélységig is lehet kódolni (a CMYK-t nem tudja kezelni).

TIF = Az ún. TIFF = Tagged Image File Format nagyon elterjedten használt, főleg a DTP területén. Fontos jellemzője az operációs rendszer és a hardverfüggetlenség.

JPG = A veszteséges képtömörítési szabvány fájlformátum.

FIF = Veszteséges fraktáltömörítéses fájlformátum, elérhető kb. 1:100 arány is minőségromlás nélkül.

GIF = A weben legelterjedtebb veszteségmentes tömörítésű fájlok formátuma.

PDF = Az Adobe cég által kifejlesztett fájlformátum (Portable Document Format). Alapját Postscript lapleírónyelv képezi.

PCD = A digitális fényképekre a Kodak által kifejlesztett szabvány.

PNG = Portable Network Graphics, a Compuserve hálózat fájl típusa.

63. Soroljon fel néhány vektoros fájlformátumot!

EPS (Encapsulated Postscript) = Raszteres és vektoros képek nyomdai munkákhoz való előkészítése során alkalmazhatjuk.

WMF = Windows Metafile.

WRL = VRML fájlok formátuma.

DXF = Huzalvázás vektoros (CAD) objektumokkal kapcsolatos adatcserére használatos (Drawing Exchange Format).

IGES = (Initial Graphics Exchange Specifications) vektoros fájl szabvány.

3. A VEKTORGRAFIKUS MODELLEZÉS

64. Jellemezze a vektorgrafika modellterét!

A vektorgrafika modelltere a két- vagy háromdimenziós euklideszi tér. A grafikus objektumokat egy „végtelen” két- vagy háromdimenziós lebegőpontos koordináta-rendszerben ábrázoljuk, kezelésük matematikai eszköze a vektoralgebra.

65. Mi jellemzi a 3D-s vektorgrafikus modellter objektumait?

A valódi 3D-s modellterek objektumait a számítógépen belül teljes értékű háromdimenziós alakzatokként kezeljük. Ezeknek az objektumoknak előállíthatjuk különböző irányú nézeteit, ezek képernyőparancsokkal interaktív módon feldolgozhatók.

Így az olyan tipikus 3D-s műveletek, mint az eltolás, forgatás, tükrözés, másolás a képernyőn való megjelenítéstől teljesen függetlenül végrehajthatók a 3D-s modellter objektumain.

66. Mit nevezünk világkoordináta-rendszernek?

A számítógépes grafikában a koordinátákat lebegőpontos számként ábrázoló Descartes koordináta-rendszereket világkoordináta-rendszereknek nevezik.

67. Hogyan írjuk le matematikailag a térbeli görbéket?

A térbeli felületeket kétparaméteres $\underline{r} = \underline{r}(u, v)$ vektor-skalár egyenletükkel modellezhetjük, ahol

$$u \in [u_1, u_2] \text{ és } v \in [v_1, v_2] \text{ a paramétertartomány.}$$

68. Milyen egyenlettel modellezhetők a térbeli felületek?

ua.

69. Mi a koordináta-transzformáció, és mire használható a számítógépes grafikában?

Koordináta-transzformációról akkor beszélünk, ha a tárgypontok egy új koordináta-rendszerre

vonatkozó koordinátáit határozzuk meg, a régiek ismeretében. Ilyenkor tehát a vizsgált tárgy változatlan, csupán nézőpontunkat változtatjuk meg.

Koordináta-transzformációnál a grafikus objektum változatlan marad (nem torzul, nem változtatja meg az alakját). Ilyen transzformációra például akkor van szükség, ha a nézőpontunkat változtatjuk a 3D-s térben.

70. Mi a ponttranszformáció, és mire használható a számítógépes grafikában?

Ponttranszformációról akkor beszélünk, ha a grafikus objektumhoz annak valamilyen értelemben vett hasonmását rendeljük. Tipikus példa erre például a fényképezés, ahol a 3D-s tárgyak egyes pontjaihoz egy 2D-s kép pontjait rendeljük hozzá. Ide értjük továbbá a testek elforgatását, elmozgatását stb. is.

71. Mit nevezünk affin transzformációnak?

Láthattuk, hogy a koordináta-, illetve a ponttranszformációk közül leggyakrabban a számítógépes grafikában a következőket használjuk:

- Egybevágósági transzformációk:
 - eltolás,
 - forgatás,
 - centrális tükrözés.Ekkor a test képével egybevágó (méretazonos).
- Hasonlósági transzformációk:
 - kicsinyítés,
 - nagyítás.Ekkor a test képe az eredetihez képest méretarányosan változik meg. Nem változik meg viszont a test alakja és szögei.
- Általános léptékváltás:
 - összenyomás,
 - széthúzás.Ekkor a test képe különböző irányokban eltérő módon torzul.

Ezeket a transzformációkat összefoglaló néven affin transzformációknak nevezzük. Jellemzőjük többek között például a következő:

72. Mit nevezünk vetítésnek?

Vetítésnek nevezzük az olyan dimenzióvesztéssel járó ponttranszformációt, amelyeknél a tárgyponatok és a megfelelő képpontok párhuzamos vagy egy ponton átmenő egyeneseken helyezkednek el.

A tárgy- és képpontokon áthaladó egyenest vetítősugárnak nevezzük. A vetítés eredménye a vetület, ami egy térbeli síkon – a képsíkon – képződik.

73. Mit nevezünk párhuzamos és középpontos vetítésnek?

Párhuzamos vetítésről beszélünk, ha a vetítősugarak egymással párhuzamosak. Ha ezen kívül a vetítősugarak még merőlegesek is a képsíkra, akkor merőleges a vetítés, egyébként pedig a ferde vetítés elnevezést használjuk

A középpontos vagy perspektivikus vetítés esetén a vetítősugarak mindegyike áthalad egy vetítési

középponton, a centrumponton (lásd az ábra b) részét). A létrejövő kép igen közel áll az emberi szem, illetve a fényképezőgép által alkototthoz.

74. Mi a P pont normalizált homogén koordinátája, ha 3D-s koordinátái x, y, z ?

A $P(x, y, z)$ pont homogén koordinátáinak nevezzük a $(w \cdot x, w \cdot y, w \cdot z, w)$ koordináta négyest, ahol w egy tetszőleges, nemzérus skalár. Ha w értékét 1-nek választjuk, akkor az $(x, y, z, 1)$ koordináta négyest a P pont normalizált homogén koordinátáinak nevezzük.

75. Írja fel az affín transzformációk egyenletét homogén koordinátákban!

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$$

76. Mit tartalmaz a geometriai objektumokról a vektorgrafikus adatbázis?

77. Mi a különbség az adat- és eljárásorientált geometriai modellezés között?

Adatorientált esetben a térbeli alakzat jellemző adatait tároljuk a számítógépes rendszerben (például háromszög esetében a csúcspontokhoz vezető vektorok koordinátáit) míg eljárásorientált esetben a térbeli alakzat generáló programját (például körgeneráló rutin a kör egyenlete és a középpontjának és sugarának paraméterei alapján).

78. Milyen kapcsolatok lehetnek a geometriai objektumok között?

- alá-fölérendeltségi hierarchikus viszonyok, amelyek jellemző változata a tartalmazás (például: ház \rightarrow tetőszerkezet \rightarrow tetőablak),
- mellérendeltségi viszonyok.

79. Milyen tulajdonságokat rendelhetünk a vektorgrafikus geometriai objektumokhoz?

A vektorgrafikus adatbázis geometriai objektumaihoz különböző – jellemzően megjelenítésorientált – tulajdonságokat is hozzárendelhetünk. Ilyenek például a rasztergrafikához hasonlóan

- a szín,
- a vonalstílus,
- a felületi jellemzők: textúrák, felületre vetített raszteres képek, érdekesség stb.,
- a szövegek.

80. Milyen adatokat tartalmaz a vektorgrafikus adatbázis?

- 3D-s világkoordináta-rendszerben meghatározott geometriai objektumok adatait:
 - o a modell neve, azonosítói, a geometriai alakzatot felépítő geometriai építőelemek fajtái,
 - o az építőelemek kapcsolódásaira vonatkozó adatok,
 - o a geometriai alakzatra vonatkozó méret-, nagyságadatok,

- a geometriai alakzatra vonatkozó hely- és helyzetadatok a modellezési világkoordináta-rendszerben,
 - a geometriai alakzat tulajdonság adatai,
 - a geometriai alakzat megjelenítésének adatai.
- a geometriai objektumok közötti viszonyokat meghatározó strukturális kapcsolatok adatait,
 - a modell tér geometriai objektumaihoz rendelt mennyiségi és szervezési információk adatait.

81. Hogyan lehet megoldani a modellezési algoritmusok egységes kezelését?

A számítógépes grafikában ezekre figyelemmel dolgozták ki a különböző görbe- és felületmodellezési algoritmusokat. Ezeknek tehát meg kell felelniük a következő gyakorlati követelményeknek:

- az algoritmusokat programtechnikailag egységesen, relatíve nem nagy ráfordítással kell megvalósítani. Ezért olyan modellezési eljárások kellene, amelyek kizárják az egyedi, egy-egy speciális függvényhez kötődő megoldásokat. Ugyanakkor a modellezési eljárásnak a legkülönbözőbb formájú, alakú görbéket, felületeket is elő kell tudnia állítani,
- a modellezési eljárás biztosítsa az előre megadott pontokon áthaladó térgörbék és felületek hatékony generálását, támogassa a különböző térelemek illeszkedésének (metszéspontok, érintés) kezelését,
- a modellezési eljárásban felhasznált függvényeknek viszonylag egyszerűen kiszámíthatóknak kell lenniük. Az egységes programtechnikai kezelés miatt ezek azonos tulajdonságú függvénycsaládokból kerüljenek ki,

82. Milyen illeszkedési feladatokat kell megoldani a görbék és felületek modellezés eljárásával?

83. Hogyan biztosítható a modellezés során felhasznált függvények egyszerű kiszámíthatósága?

ua.

84. Sorolja fel, milyen követelményeket kell kielégíteni a vektorgrafikus modellezésnek a felhasználóval való interaktív kapcsolattartás végett!

A felhasználóval való hatékony, interaktív kapcsolattartás érdekében

- a modellezési eljárás biztosítsa, hogy a felületek és görbék transzformálása (például térbeli mozgítás, vagy a paramétertartomány lineáris transzformációi) relatíve egyszerű algoritmussal, nem túl nagy számításigénnyel megvalósítható legyen. (Például ne legyen szükség nagy számú görbepont transzformált vektorainak egyenkénti kiszámítására.) Másként fogalmazva, a modellezési eljárásnak invariánsnak kell lennie az affin transzformációkra és a vetítésekre,
- a modell olyan legyen, hogy a képernyőn való megjelenítéshez szükséges poligon és sokszöglapos közelítésre gyorsan konvergáló algoritmusokat lehessen kidolgozni,
- a modell tegye lehetővé a görbék és felületek lokális változtathatóságát, azaz ha például egy pontban kissé megváltoztatjuk a görbét, ez csak a pont közvetlen környezetében okozzon változást (lásd az ábrát).

85. Milyen fokszámú polinomokat célszerű választani a modellezéshez, és miért?

Nyilvánvaló, minél kisebb a polinom fokszáma, annál kevesebb művelettel számíthatjuk ki a függvényértékeket. Viszont túl alacsony fokszámú polinomot sem választhatunk, mivel akkor a „bonyolultabb” görbéket és felületeket nem tudnánk jól közelíteni.

Előbbiek miatt a számítógépes grafikában a térgörbék és felületek modellezésére a harmadfokú polinomokat választották.

86. Mivel közelítjük a térgörbét és a felületeket a számítógépes grafikában?

Tehát a térgörbét az

$$\underline{r} = \underline{a}_0 + \underline{a}_1 \cdot t + \underline{a}_2 \cdot t^2 + \underline{a}_3 \cdot t^3$$

alakú köbös paraméteres ívekkel, a felületeket pedig kétparaméteres köbös felületekkel közelítjük

87. Milyen két eltérő módszer létezik az interpolációs és approximációs feladat megoldására?

- az összes interpolálandó vagy approximálandó pontot figyelembe véve egyetlen görbét vagy felületet határozunk meg,
- az interpoláló vagy approximáló görbét, illetve felületet egymáshoz folytonosan kapcsolódó részekből állítjuk össze.

88. Mit nevezünk spline görbéknek?

Ha a modellezendő térgörbét több, egymáshoz folytonosan kapcsolódó ívből állítjuk elő, akkor ezt a görbét spline-nak nevezzük.

89. Mit nevezünk spline felületeknek?

Ha egy modellezendő felületet részekből állítunk össze, akkor ezt spline felületnek nevezzük. A spline felületek részeit leggyakrabban köbös polinomokkal generáljuk, ezeket két paraméteres köbös felületfoltoknak (bicubic patch) nevezzük.

90. Miért előnyös egy kontrollpontokkal generált görbe esetében, ha invariáns az affin transzformációkra?

91. Definiálja a Bézier-íveket!

A Bézier approximációs ívekkel történő modellezést különösen az autógyártás CAD rendszereiben alkalmazzák, mivel ezzel a módszerrel a megvilágított karosszéria törésmentes fényjátéka viszonylag jól visszaadható.

A harmadfokú Bézier-íveket [jelölése $\underline{B}_3(t)$] a következő képlettel definiálhatjuk, ha adott a görbe P_0, P_1, P_2, P_3 kontrollpontja az $\underline{r}_0, \underline{r}_1, \underline{r}_2$ és \underline{r}_3 vektorokkal:

$$\underline{B}_3(t) = \sum_{j=0}^3 \underline{r}_j \cdot B_j^3(t) \quad t \in [0,1]$$

$$\text{ahol } B_i^3(t) = \binom{3}{i} \cdot t^i (1-t)^{3-i} \quad i = 0, 1, 2, 3 \quad 0 \leq t \leq 1$$

92. Milyen tulajdonságai vannak a Bézier-íveknek?

A Bézier-ívek néhány, a számítógépes grafikában fontos tulajdonsága:

- A görbe íve mindig a P_0, P_1, P_2, P_3 kontrollpontok által meghatározott négyszög belsejében helyezkedik el.
- A Bézier-ívek kontrollpontjaik affin transzformációjával szemben invariánsak. Ez azt jelenti, hogy például a görbe mozgatasakor elegendő a kontrollpontokat transzformálni, amellyel igen sok számítás

megtakarítható. (Ez azonban csak az affín transzformációkra igaz, és nem minden esetben teljesül a projektív transzformációkra, például a centrális vetítésre.)

- A Bézier-ívek globálisan változtathatók, azaz ha a kontrollpontok közül egyet elmozgatunk, az az egész görbére kihat.

- Egy egyenes pontosan annyi pontban metszi a Bézier-ívet, ahány metszéspontja van a tartónégyszögével (lásd az ábrát) (variation diminishing property).

93. Mire használható a de Casteljau-algoritmus?

A Bézier-ívek megjelenítése szempontjából nagyon fontos, hogy létezik-e olyan algoritmus, amely alapján az ív egyenesszakaszokkal közelíthető. Többek között erre a problémára ad választ a de Casteljau-algoritmus. Ez egy rekurzív eljárást ad a Bézier-ív egy konkrét t paraméterértékhez tartozó pontjának kiszámítására és az ív egyenesszakaszokkal való közelítésére.

94. Milyen részekből épülnek fel a B-spline görbék?

Matematikailag bizonyítható, hogy a B-spline görbék rendelkeznek a Bézier-görbék összes előnyös tulajdonságával és ezen túlmenően lokálisan is változtathatók, azaz egy kontrollpont elmozgatása csak a neki megfelelő $[t_i, t_{i+k}]$ intervallumban módosítja a görbe alakját.

95. Definiálja a B-spline görbéket! Mi a B-spline görbe súlyfüggvénye és Boor-poligonja?

A B-spline görbék ugyanúgy, mint a Bézier-görbék, közelíthetők poligonokkal a kontrollpoligon szakaszainak rekurzív felosztásával, és ez az algoritmus elő is állítja a B-spline görbe egy pontját egy konkrét t értékre. Ekkor a közelítő sokszögek a görbéhez tartanak határértékben. Ezzel egy hatékony eljárást kapunk a B-spline-ok raszteres képernyőn való megjelenítésére. Ezt az eljárást Cox–de Boor-algoritmusnak nevezzük, amely a de Casteljau-algoritmus általánosítása.

96. Miben különböznek a B-spline és a Bézier-görbék?

97. Mi a Cox–de Boor-algoritmus lényege?

98. Miért vezették be a racionális görbéket a számítógépes grafikában?

A B-spline görbék már majdnem kielégítik a modellezéssel kapcsolatos összes követelményt, de még mindig nem rendelkeznek a centrális vetítésekre vonatkozó invariancia tulajdonságával. Ezért vezették be a racionális görbéket a számítógépes grafikában, amelyeket először S. A. Coons alkalmazott.

99. Minek a rövidítése a NURBS és mit jelent?

A racionális B-spline-ok közül a legfontosabbak a nem uniform (azaz nem egyenközű $[t_i, t_{i+1}]$ felosztással kapott) racionális B-spline görbék, amelyeket NURBS-nek szoktak rövidíteni (NURBS = Non Uniform Rational B-Spline).

100. Melyek a NURBS legfontosabb tulajdonságai?

- A NURBS invariáns az affín transzformációkra és a vetítésekre (projektív transzformációkra). Ez azt jelenti, hogy például a NURBS görbe centrális vetítésekor elegendő a Boor-pontokat homogén koordinátákban transzformálni, és ezekből kiszámítani a vetületi görbét.

- Egy kontrollpont megváltoztatása csak a megfelelő paraméter tartományhoz tartozó pontokban módosítja a görbét, azaz a NURBS lokálisan változtatható.

- A NURB-s görbe a kontrollpoligon belsejében (konvex burkában) helyezkedik el.
- Ha a kontrollpontok egy egyenesen helyezkednek el, akkor a NURBS egyenesszakasz.
- Ha az egyik r_i kontrollpont w_i súlyát megnöveljük, a görbe határértékben a P_i ponthoz tart

101. Mit nevezünk vonalfelületnek?

Ha egy felület bármely pontján át húzható olyan egyenes, amelynek pontjai a felülethez tartoznak, akkor ezt a felületet vonalfelületnek nevezzük.

102. Hogyan hozhatunk létre felületeket görbék mozgatásával?

Felületeket létrehozhatunk úgy is, hogy egy adott generáló térgörbét önmagával párhuzamosan mozgatunk egy vezérgörbe mentén (látható az ábrán).

103. Határozza meg a Bézier-felületeket!

Bézier-felületnek nevezzük azokat a felületeket, amelyek Bézier-görbék mozgatásával úgy jönnek létre, hogy a mozgatott görbe kontrollpontjai szintén Bézier-görbéken mozognak.

104. Határozza meg a B-spline felületeket!

B-spline felületnek nevezzük azokat a felületeket, melyek B-spline görbék mozgatásával úgy jönnek létre, hogy a mozgatott görbe kontrollpontjai szintén B-spline görbéken mozognak.

105. Definiálja a NURBS felületeket!

A NURBS felületeket teljesen analóg módon származtatjuk a Bézier és B-spline felületekkel. Ekkor egy racionális, nem uniform B-spline görbét mozgatunk úgy, hogy kontrollpontjai szintén NURBS görbéken mozogjanak.

106. Mit nevezünk huzalvázmodellnek?

A huzalvázmodell a 3D-s geometriai alakzatot csúcsaival és éleivel jellemzi, ennek megfelelően a modell csak a csúcsoakat és az ezekhez rendelt összekötő éleket tartalmazza (lásd az ábrákat).

107. Mi az előnye a huzalvázmodell alkalmazásának?

A huzalvázmodellek legnagyobb előnye, hogy számítógépes megvalósításuk algoritmusigénye a többi geometriai modellezőmodszernél lényegesen kisebb, így viszonylag kis erőforrású számítógépen is használhatók.

108. Milyen korlátai vannak a huzalvázmodell alkalmazásának?

A huzalvázmodellek legnagyobb problémája, hogy egy huzalvázmodellnek több test is megfelelhet. Nem mindig tehető különbség a tömör és üreges test között a modell alapján, és a testet határoló felületek görbültségét sem tudjuk kezelni.

109. Mit tárolunk az adatbázisban huzalvázmodell esetén?

A huzalvázmodell adatstruktúrájának lényege a csúcs-, az él- és az él-csúcs táblázatok együttese, amelyeket a relációs adatbáziskezelés szabályainak megfelelően építenek fel.

110. Hogyan jellemezzük a geometriai objektumokat palástmodellezésnél? Mi a b-rep?

A palástmodellezésnél a geometriai objektumokat a vektorgrafikus modell térben határolófelületeikkel (beleértve e felületek csatlakoztatására vonatkozó adatokat is) jellemezzük.

Ennek a modellezési módszernek a neve az angol szakirodalomban boundary-representation vagy röviden

b-rep.

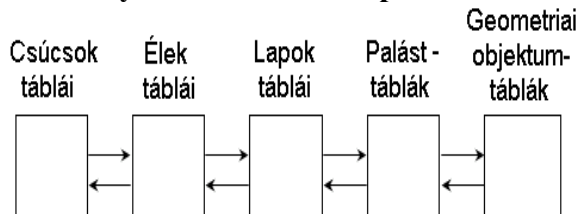
111. Mit nevezünk lépésenkénti szerkesztésnek?

Palástmodelleket létrehozhatunk lépésenkénti szerkesztéssel. Ekkor a test határoló felületeit egyenként definiáljuk a térbeli felületek csatlakoztatási lehetőségeinek függvényében.

112. Mivel generáljuk a testeket határoló felületeket valósághű palástmodellezésnél?

A műszaki tervezőrendszerek fejlettebb formáinál alkalmazott „valósághű palástmodellezés” a testet behatároló felületfoltok generálása és illesztése során a B-spline, a Bézier és NURBS felületfoltok alkalmazását is lehetővé teszi.

113. Milyen táblákból áll a palástmodell adatbázisa?



114. Milyen két típusa van a tömör testmodellezésnek?

- elemi testekkel és ezek közötti szabályos halmazműveletekkel való modellezés és
- a testek elemi sejtekből való felépítése.

115. Mi a CSG modellezés lényege?

A véges számú tömör elemi testprimitívből kiinduló és a modellt a metszet, egyesítés, kivonás és ragasztás halmazműveletek egymás utáni felhasználásával megkonstruáló modellezési módszert konstruktív tömör testmodellezésnek nevezzük. Ennek angol elnevezése: Constructive Solid Geometry, vagy röviden CSG.

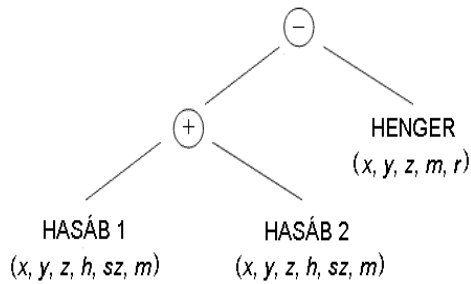
116. Milyen standard testprimitívek vannak a CSG-ben?

- a hasáb,
- a gúla,
- a henger,
- a kúp és
- a gömb

117. Milyen műveletek alkalmazhatók a CSG-ben?

- konkrét primitívpéldány létrehozása,
- objektum másolása, törlése, transzformálása,
- objektumok halmazalgebrai metszete, egyesítése, kivonása, összeragasztása.

118. Milyen a CSG modellek adatstruktúrája?



Jelölések:

x, y, z = helykoordináták;

h = hossz;

sz = szélesség;

m = magasság;

r = körsugár.

119. Miért használható a tömör testmodellezés a műszaki tervezésben?

A konstruktív tömör testmodellezésnek a gyakorlatban történő jól használhatóságára hívják fel a figyelmet azok a vizsgálatok, amelyek eredményei szerint a műszaki tervezés során szükséges testek döntő része előállítható néhány egyszerű geometriai test (például hasáb és henger) megfelelő kombinációjából.

120. Mi a térfogatmodellezés alapelve? Mit nevezünk voxelnek?

A térfogatmodellezésnél (cell modeling) egy tömör tárgyat több egymáshoz csatlakozó, de egymást nem metsző kisebb tömör tárgyra, azaz sejtekre bontunk fel. Az elterjedtebb modellezési módszerek a sejtek két típusát kezelik:

- a sejtek azonos típusú alakzatok (például hasábok), de méretük egy paramétertől függően változhat,
- a sejtek azonos típusú és méretű alakzatok, ekkor ezeket (a képponthoz, azaz a pixelhez hasonlóan) voxelnek (volumen element) nevezzük.

121. Hogyan modellezhetjük a testeket voxelekkel?

Az azonos formátumú és méretű voxelekkel való kitöltése a modellezendő testnek, ha a voxeleket elegendő kis méretűre választjuk, a test relatíve pontos leírását eredményezi. A leggyakoribb voxeltípus a kocka. A modellezendő objektumokat a voxelekkel úgy írjuk le, hogy minden egyes, a testhez teljes egészében vagy csak részlegesen hozzátartozó voxel adatait hozzárendeljük a testhez.

122. Mit nevezünk sugárzási teljesítménynek? Mi a lumen?

$$\varphi = \frac{dQ}{dt}$$

menyiséget, azaz a fényforrás által időegység alatt kisugárzott

energiát sugárzási teljesítménynek vagy fluxusnak nevezzük, egysége a watt [W].

A sugárzási teljesítmény fotometriai megfelelője a fényáram, amelynek mértékegysége a lumen [lm].

123. Mit nevezünk kisugárzás-erősségnek? Mi a candela?

Ha a pontszerű fényforrás az egységsugarú gömb egységnyi térszögébe időegység alatt I energiát sugároz ki, akkor I -t a kisugárzás-erősségnek nevezzük (radiant intensity). A definícióból következőn

$$\frac{d\phi}{I=d\Omega}$$

Az I sugárzás-erősség megfelelője a fotometriában a fényerősség. Ennek egysége a candela [cd], ami kb. egy gyertya fényének felel meg.

124. Mi a besugárzás-erősség?

Az egységnyi felületre időegység alatt eső sugárzási energiát besugárzás-erősségnek (irradiance) nevezzük:

125. Mit mond ki a Lambert-féle távolság- és koszinusztörvény?

A természetes fényforrások nem minden irányban sugározzák be azonos erősséggel a felületet, a felületre eső sugárzás-erősség a távolságtól is függ. Az előbbi képletből, figyelembe véve,

hogyan $d\Omega = I \frac{dA \cos \alpha}{R^2}$, megkapjuk a Lambert-féle négyzetes távolság és koszinusz törvényt:

$$E = I \frac{\cos \alpha}{R^2}$$

ahol R a felület távolsága a pontszerű fényforrástól, α a felület normálvektorának és a fényforrástól a felületig vezető vektor (sugárzás iránya) által bezárt szög.

126. Mi a megvilágítás-erősség mértékegysége?

A besugárzás-erősségnek a fotometriai megfelelője a megvilágítás-erősség, amelynek egysége a lux [lx].

127. Definíálja a sugársűrűséget és a fénysűrűséget!

Sugársűrűségnek (radiance) nevezzük azt az L -vel jelölt mennyiséget, amely az egységnyi térszögből időegység alatt az egységnyi kisugárzó felület vizsgált irányú vetületéről származó sugárzási energia. Felületek által adott irányban kisugárzott teljesítmény jellemezhető ezzel a mennyiséggel.

A sugársűrűség fotometriai megfelelője a fénysűrűség.

128. Mi jellemzi a Lambert-fényforrásokat?

Azokat a testeket, amelyek oly módon sugároznak ki fényt, hogy a fénysűrűség irányfüggetlen, Lambert-féle fényforrásoknak nevezzük.

129. Milyen típusú fényforrásokat adhatunk meg a vektorgrafikus rendszerekben?

- szórt háttérvilágítás, amelynek nincs iránya,
- távoli fényforrások, amelyek irányított párhuzamos fénysugarakat bocsájtanak ki,
- pontszerű fényforrások, amelyek egy térbeli pontból minden irányban kibocsájtanak fénysugarakat,
- reflektor-fényforrások, amelyek a tér egy adott helyétől egy meghatározott irányban, párhuzamos vagy

csonka kúp alakú fénynyalábot sugároznak,
- kiterjedt fényforrások, amelyek meghatározott méretű és alakú sugárzó testek.

130. Mit jelent a szórt háttérvilágítás (ambient light)?

A szórt háttérvilágítás hatása jó közelítéssel a nappali fényviszonyoknak felel meg egy erősen felhős égbolt esetén, amikor a testek egyenletesen, minden irányból kapnak fényt. A szórt háttérvilágítás (ambient light) esetében tehát a modelltér összes objektuma a tér minden irányából azonos erősségű fénnel lesz megvilágítva.

131. Jellemezze a pontszerű fényforrásokat!

A pontszerű fényforrás (point light) a modelltér minden irányában azonos intenzitással sugároz. A közeli tárgyak a távolabbiaknál erősebben meg lesznek világítva, a Lambert-féle távolságtörvénynek megfelelően.

132. Mit nevezünk area light-nak?

A kiterjedt fényforrások (area light) olyan fényforrások, amelyek sugárzását geometriai alakjuk is befolyásolja. Ezeket például sugárzó síklapok esetében pontszerű fényforrások együttesével közelíthetjük.

133. Mit nevezünk fraktálnak?

A fraktál egy olyan geometriai alakzat, amelynek részei hasonlítanak az egész alakzathoz és a részek a még kisebb részekhez. Egy geometriai görbe vagy felszín tehát akkor fraktál, ha bármely részét felnagyítva az eredetivel azonos alapmotívumú, azaz önhasonló görbét vagy felszínt kapunk.

134. Mit nevezünk véletlen (sztochasztikus) fraktálnak?

Véletlen iterációs algoritmusokkal is generálhatunk fraktálokat, ezeket sztochasztikus fraktáloknak nevezzük.

135. Jellemezze a fraktálok felhasználási területeit a számítógépes grafikában!

- Természethű képek generálása véletlen fraktálokkal (rét, erdő, hegységek, felhők stb.).
- Fraktálok felhasználása díszítőelemként. Különösen a weblapok háttérének fraktálokkal való felépítése terjed rohamosan (lásd az ábrát).
- A fraktálok igen hatékonyan alkalmazhatók képtömörítésre

136. Mit nevezünk karakternek és karaktermodellezésnek?

A számítógépes grafikában az animációkészítés során a modelltérben létrehozott mozgó és alakjukat meghatározott törvényszerűségek szerint változtató, ugyanakkor képi jellegzetességük lényegét megőrző 3D objektumokat karaktereknek (character), az ezek létrehozásával és mozgásával összefüggő eljárásokat és módszereket pedig karaktermodellezésnek nevezzük.

137. Milyen módszerek vannak a karakterek létrehozására a 3D modelltérben?

A felületek előállítását történhet 3D primitívek kontrollpontjainak módosításával, 3D kihúzással (extruding), 3D pásztázással (lofting), vagy mintakarakter 3D beszkenyelésével.

138. Hogyan készítünk a 3D-ben beszkenelt makettből a grafikus rendszerben kezelhető objektumot?

A háromdimenziós szkennert a letapogatott tárgyról olyan 3D modellt készít, amely több millió referenciapontból áll. Ezek a referenciapontok határozzák meg a tárgy alakját, körvonalát. A keletkezett pontfelhőből poligonokat készítünk, így kapunk egy könnyebben kezelhető objektumot. A hatékony munkavégzéshez még ez a felbontás is magas, ezért a poligonokat NURBS felületfoltokkal közelítjük (kb. 80–100 felületfolt elegendő egy modell megépítéséhez, persze ez az érték függ a modell részletességétől is).

139. Mi a különbség a bump mapping és a displacement mapping eljárás között?

- A bump mapping eljárásnál a megjelenítő szoftver egy fekete-fehér intenzitású mintázatot figyelembe véve módosítja a felületi normál-vektorokat és ezzel kiemelkedéseket és besüllyedéseket szimulál. Ha ezt alkalmazzuk, akkor a modell térbeli objektum nem változik.
- A NURBS felületek parametrizálhatósága lehetővé teszi, hogy a felületelemeket ténylegesen eltoljuk a normál vektora irányába (displacement mapping). Ez az eljárás általában jobb eredményeket szolgáltat a bump mapping eljárásnál, de sokkal számításigényesebb és megváltoztatja a modell térbeli objektumot is.

140. Mi a feladata a karaktermodellezésben a csontváznak?

Ahhoz, hogy ezt a felszínt mozgatni tudjuk, egy olyan struktúrára van szükség, amely kisszámú vezérlő pont segítségével az összes elvárt deformációt létre tudja hozni. Ezt a problémát oldják meg az ún. kinematikus rendszerek, amelyekben a karakter részeinek mozgatása, deformálása (az emberi, állati testhez hasonlóan) egy csontvázrendszer segítségével történik.

141. Mit nevezünk kinematikai láncnak?

A csontváz (skeleton) olyan részeit, amelyekhez tartozó csontok kapcsolódnak egymáshoz és együtt mozognak, kinematikai láncnak nevezzük.

142. Mi jellemzi a forward kinematikát?

A forward kinematikában (forward kinematic) a kinematikus lánc minden egyes tagját (bone) egyenként forgathatjuk a kapcsolódási pontjai (joint) körül. Amikor egy kívánt pózba szeretnénk beállítani a karakterünket, akkor a hierarchia legtetjétől (root) elindulva forgatjuk a csontrendszert – ha szükséges egyenként – a megfelelő helyre.

143. Mi jellemzi az inverz kinematikát?

Az inverz kinematika (inverse kinematic) alap gondolata az, hogy a kinematikai láncban különböztessünk meg egy vezérlő csontrészt, amelynek mozgatása esetén a kinematikai rendszer képes a szülő rész megfelelő elmozdulását automatikusan kiszámítani.

144. Mit kell elvégezni a karaktermodellezés „beborítás” munkafázisában?

A beborítás vagy bőrizés (enveloping, skinning) lépésében a létrehozott karakter felületét alkotó elemeket hozzárendeljük a csontvázrendszerhez, figyelembe véve a mozgással járó felületi deformációkat (pl. embernél izomzat mozgása).

Két fő csoportra oszthatók a folyamatban résztvevő objektumok:

- A deformálható elemek csoportja, vagyis a “bőr”. (A bőrfelület jelen esetben nemcsak a fizikai értelemben vett bőrt jelentheti, hanem ennek szerepét sok más, a modellhez tartozó kiegészítő objektum is átveheti. Például: kesztyű, cipő stb., amelyek képesek lesznek a karakterrel együtt az alakváltozásra.)

- A deformáló objektumok csoportja, ide tartoznak a csontvázak és a modellező programok által felkínált különböző deformáló eszközök.

145. Mit nevezünk elsődleges és másodlagos mozgásnak a karakteranimációban?

A karakterek számítógépes animációja (animating) két fő részre osztható: elsődleges és másodlagos mozgásra. Az elsődleges mozgás adja meg a karakter fő mozgásait és cselekvéseit, a másodlagos pedig ezek esetleges mellékhatásait, mint például hús, bőr és izomzat deformációit.

146. Mi a fázisanimáció (key framing) lényege?

Ehhez hasonlóan a fázisrajzolók szerepét a grafikus rendszer veszi át, így a filmgyártásban elterjedt programcsomagoknak csak a karakter kulcspozícióit kell megadni, a közbeeső mozgáshoz tartozó karakterpozíciókat a szoftver már automatikusan, ún. mozgás interpolációval generálja.

147. Hogyan készítik el a karakter kulcspozícióit a 3D mozgásrögzítés (motion capture) módszerével?

Az emberi mozgás jellemzőinek rögzítésére sok technikai megoldást dolgoztak ki. Többségük közös abban, hogy a testre aktív vagy passzív érzékelőket rögzítenek, amelyek pozícióját a felvétel során nagy pontossággal mérik. Ezeknek a méréseknek az eredményei alapján azután “könnyen” kiszámítható a szereplő csontvázának helyzete és például csuklóinak pontos orientációja.

148. Mire használjuk a klipeket nemlineáris animációnál?

A nemlineáris animációkat (nonlinear animation) ennek az elvnek megfelelően az egyes mozdulatsorozatoknak megfelelő újrafelhasználható (elementett) részanimációkból, az ún. klipekből állítjuk össze.

149. Ismertesse az arcanimáció (facial animation) lépéseit!

Az arc animációját a következő lépésekben végezhetjük el:

- Lemodellezzük a karakter arcát. Ez általában kifejezéstelen “pókerarc”. Ezt nevezzük bázisobjektumnak.
- Másolatokat készítünk a bázisobjektumról.
- A másolatokat úgy deformáljuk, hogy különböző arckifejezéseket mutassanak (célobjektumok).
- Interpolációval elvégezzük a bázis- és célobjektumok közötti közbeeső állapotok generálását.

150. Hogyan lehet a vektorgrafikus objektumok helyzetére vonatkozó információkat közölni a vektorgrafikus rendszerrel?

Ha a felhasználó az alakzatok helyzetére vonatkozó információkat akar közölni a vektorgrafikus rendszerrel, akkor ezt a következő formákban teheti meg:

- a konkrét koordinátaértékek begépelésével billentyűzetről,
- a grafikus kurzor helyzetének megfelelő koordinátaértékekkel, például az egér vagy más relatív koordinátás eszköz használatával,
- abszolútkoordinátás eszköz, például digitalizáló tábla (tablet) használatával,
- a modell térben már definiált helyzetű objektumokra, vagy a képernyőn látható, a világkoordináta-rendszerben értelmezett segédeszközökre (vonalzók, segédvonalháló = grid) való hivatkozással.

151. Hogyan lehet egérrel kijelölni egy vektorgrafikus objektumot?

az egérrel a képernyőn rá kell mutatni a kijelölendő objektum képének határoló görbéjére vagy

zárt alakzatok belsejére. A grafikus kurzor rászteres képernyőkoordinátáit ekkor először át kell konvertálni a modelltérbe és meg kell állapítani, hogy a konvertált koordinátaérték melyik grafikus objektumhoz tartozik.

152. Milyen műveleteket végezhetünk a modelltér objektumaival?

- új objektum létrehozása,
- egy létező objektum transzformálása, másolása, törlése,
- meglévő objektumokkal végzett halmazalgebrai műveletek,
- struktúra képzés,
- jelenetek megjelenítése.

153. Hogyan hozunk létre primitívekből geometriai objektumokat a modelltérben? Milyen 2D-s és 3D-s primitíveket ismer?

a geometriai alakzatokat a primitíveknek a modelltérben való „összeépítésével” hozzuk létre.

A 2D-s rajzolóprogramok az alakzatokat általában

- a vonal (szakasz),
- a téglalap (négyzet),
- az ellipszis (kör),
- a sokszög

primitívek véges számú kombinációjával állítják elő.

A 3D-s vektorgrafikus rendszerekben ezek általában még a következő térbeli primitívekkel egészülnek ki:

- hasáb (beleértve a téglatestet és a kockát is),
- gúla (csonka gúla),
- henger,
- kúp (csonka kúp),
- gömb,
- törusz.

154. Mit jelent a szerkesztéssel történő objektumdefiniálás?

Ebben az esetben a felhasználó adja meg a vektorgrafikus rendszer számára az összes információt, amely alapján a térbeli test összeállítható. Ez a módszer tipikusan jellemző a testek palástfelületekkel való modellezésére (b-rep), amikor a felhasználó egyenként meghatározza az egyes fedőlapok, felületek jellemzőit és az ezek csatlakoztatására vonatkozó adatokat.

155. Mit jelent a pásztázás?

A 3D-s vektorgrafikus objektumok generálásának fontos módszere a pásztázás (ezt gyakran söprésnek is nevezik, angolul sweep). Pásztázásnál egy 2D-s felületelemet mozgatunk egy vezérgörbe mentén, és ennek során a felületelem által „súrolt” térbeli pontok egy testet határoznak meg.

156. Mit jelent a kihúzás?

A kihúzásnál a síkra merőlegesen, egy egyenes mentén „kihúzzuk” a 2D-s alakzatokat (lásd felső ábra). Speciális esetének tekinthető az, amikor egy 2D-s rászteres kép szűrkeség értékeit magasság koordinátákként értelmeztük (height field, lásd alsó ábra).

157. Hogyan hozhatunk létre térbeli alakzatokat forgatással?

Forgatásnál a vezérgörbe kör, a generáló alakzat síkja a forgatás bármely pillanatában merőleges a

vezérgörbére, és a generáló alakzat pontjai is körön mozognak. A példát az ábra mutatja.

158. Mit jelent a vektorgrafikus objektum másolása?

A másolás a kijelölt vektorgrafikus objektum egy eltérő nevű (azonosítójú) példányát hozza létre a világkoordináta-rendszer egy másik helyén.

159. Milyen transzformációkat tesznek lehetővé a vektorgrafikus rendszerek?

- az eltolását,
- az elforgatását (ide értve a tükrözést is) és
- a léptékváltást (nagyítás, kicsinyítés, összenyomás, széthúzás).

160. Melyek a vektorgrafikus objektumokkal végrehajtható Boole-algebrai műveletek?

A napi gyakorlati munkavégzés komfortjához ezek mellett még szükség van az összeragasztás (GLUE) műveletére is

161. Mi a hatása az összeragasztás műveletének?

a műveletben résztvevő testeket egy közös felületen csatlakoztatjuk

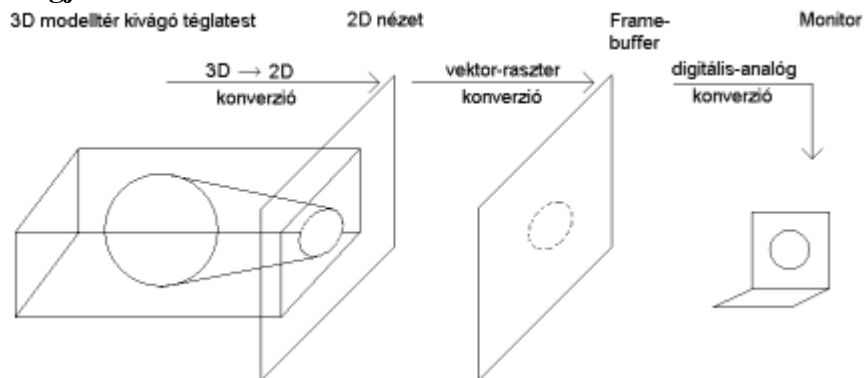
162. Mi a lényege a fóliák alkalmazásának?

hogy az objektumok a felhasználó által megadott csoportosításban külön fóliákra kerülhetnek, amelyek a rendszerben külön kezelhetők. Ezt a 2D-s rajzelemek esetében úgy képzelhetjük el, hogy a rajz egyes részeit különálló, teljesen átlátszó pauszra készítjük el. Ezeket külön-külön is megtekinthetjük. Egyémásra rakva az átlátszó fóliákat, láthatjuk a teljes rajzot is.

Az egyes fóliák „lefagyaszthatók”, ekkor tartalmuk például a képernyőn nem jelenik meg. Amennyiben viszont később a lefagyasztott fóliára ismét szükség lenne, ezeket „elővehetjük”, és így többcélúan is felhasználhatjuk.

4. A VEKTORGRAFIKUS MEGJELENÍTÉS

163. Milyen lépések szükségesek a vektorgrafikus objektumok képének a monitor képernyőjén való megjelenítéséhez?



164. Mit nevezünk jelenetnek (scene)?

A modelltérnek egy adott nézőpontból látható és a képi megjelenítés szempontjából összetartozó

objektumait jelenetnek (scene) nevezzük.

165. Mit adunk meg a KAMERA paranccsal?

A vektorgrafikus rendszerekben egy jelenet képének meghatározásához szükséges nézőpontot és irányt a világkoordináta-rendszerben általában a KAMERA paranccsal definiáljuk.

166. Hogyan lehet a vektorgrafikában megvilágítani a modelltér objektumait?

Ezt megtehetjük például azzal, hogy a világkoordináta-rendszerben meghatározott helyű, erősségű és színű fényforrásokat definiálunk.

167. Mit jelent a renderelés?

A modelltér egy jelenetéből a raszteres kép előállítását renderelésnek (rendering) nevezzük. A fogalom a latin reddere = áthelyez, fordít, átalakít szóból származik, és ilyen értelemben a modelltér objektumainak digitális képpé való átalakítását jelenti.

168. Milyen tipikus formái vannak a vektorgrafikus objektumok megjelenítésének?

WIREFRAME = huzalvázás megjelenítés

NET = poliéder közelítésű vonalhálós megjelenítés

HIDDEN = takartvonalas palástmodell

SHADE = felületárnyalt testmodell

169. Mi a huzalvázás képi megjelenítés lényege?

A vektorgrafikus objektumok huzalvázás vagy drótvázás (wireframe) megjelenítése a legegyszerűbb, ugyanakkor a legkevésbé valóságos. Ez esetben a testeket éleikkel ábrázoljuk, azaz a testeket a képen úgy jelenítjük meg, mintha drótból készültek volna.

170. Mit nevezünk árnyalt megjelenítésnek?

Ekkor a képernyőre nem az objektumok átlátszó vázát, hanem határoló felületeik színnel vagy felületi mintázattal kitöltött képét rajzoljuk ki.

171. Mit jelent a fotorealisztikus megjelenítés?

Fotorealisztikus megjelenítés alatt a számítógépes grafikában azt értjük, hogy a modelltérbeli jelenetről olyan minőségű képet állítunk elő egy vektorgrafikus rendszerrel, amely teljesen valószerű és a valós világról készített fényképtől nem különböztethető meg.

172. Milyen követelményeknek kell megfelelnie egy 2D-s képnek, hogy térhatású legyen?

A fotorealisztikus megjelenítés egyik fontos követelménye, hogy a 3D-s modelltér jelenete a 2D-s raszteres képen is térhatású legyen.

173. A felületeknek milyen tulajdonságait kell figyelembe venni realisztikus képek készítésekor?

- A nézőponttól távolodva a képen a párhuzamos egyeneseknek fokozatosan összetartóknak kell lenniük, és a távolabbi testek méreteit arányosan csökkenteni kell. Ezeket a hatásokat a megjelenítés során megfelelő perspektíva transzformációkkal érhetjük el.
- A 2D-s képen szerepeltetett jelenet objektumai egy adott nézőpontból szemlélve takarhatják egymást. Fontos tehát, hogy a képen reálisan ábrázoljuk a tárgyak látható és nem látható éleit, felületeit.

- A nézőtől távoli objektumok már nem látszanak olyan tisztán, mint a közeleiek. Ezért a képen a „messzeségbe tűnő” tárgyaknak elmosódottabbaknak és kevésbé részletesen kidolgozottaknak kell lenniük. Ezt a hatást elérhetjük a színintenzitások megfelelő változtatásával, illetve a textúráknak a nézőponttól való távolság függvényében történő alkalmazásával (például mip-mapping).

174. Mit kell figyelembe venni az átlátszóság modellezésénél?

Az átlátszóság modellezésénél figyelembe kell vennünk a fénytörést, a diffúz áttetszőséget és a fény intenzitásának csökkenését az átlátszó testen történő áthaladás során

175. Adja meg a modellter jeleneteiből a raszteres kép előállításának lépéseit!

- Meg kell adni a modellter objektumait megvilágító fényforrásokat a világkoordináta-rendszerben.
- Rögzítenünk kell a nézőpontot vagy a kameraállást a világkoordináta-rendszerben, ahonnan a jelenetet szemléljük.
- El kell döntenünk, hogy a modellter milyen objektumait kívánjuk szerepeltetni a generálandó képen. Ehhez egy ablakot (window) kell definiálni a világkoordináta-rendszerben, amelyen keresztül a nézőpontból a jelenetet látjuk. Azok az objektumok, amelyek ezen az „ablakon” kívül esnek, nem vesznek részt a képgenerálásban, azokat kivágjuk. A megjelenítésnek ezt a lépését ablakozásnak és kivágásnak (windowing and clipping) nevezik a szakirodalomban.
- A modellternek a jelenetben szereplő objektumait a világkoordináta-rendszerből affin és perspektív transzformációval egy normalizált ábrázolási térbe kell leképeznünk.
- Az ábrázolási térben meg kell határozni az objektumok takarási viszonyait, azaz a nézőpontból látható éleket és felületeket. Ez a látható kép meghatározó algoritmusokkal történik.
- A látható felületelemek képpontjaihoz ezt követően az árnyalási algoritmusokkal hozzárendeljük a fényviszonyoknak és a textúráknak megfelelő színeket.
- A raszteres képernyőn a kiválasztott ablaknak megfelelő pixelekre „vetítjük” a felületelemek képpontjainak színértékeit a monitor fizikai eszközkoordináta-rendszerében.

176. Mit nevezünk képgenerálási pipeline-nek?

A vektorgrafikus objektumok definiálásával kezdődő és a képernyőn való megjelenítésükig tartó teljes folyamatot a szakirodalomban képgenerálási pipelinenak (viewing pipeline, vagy rendering pipeline) nevezik.

177. Mivel közelítjük a görbéket és felületeket a megjelenítéshez? Mi a tesszelláció?

A raszteres kép előállítása során tehát a térbeli görbéket mindig poligonokkal, a felületeket pedig sokszöglapokból – legtöbbször háromszögekből – felépített alakzatokkal közelítjük. A modellterbeli felületek sokszögekre bontásának folyamatát tesszellációnak (tessellation) nevezzük.

178. Hogyan definiálhatunk egy kamerát?

meg kell adni annak a pontnak a koordinátáit a világkoordináta-rendszerben, ahova a fényképezőgépet elhelyezzük. Ez azonban a pontos képalkotáshoz még kevés, szükség van a fényképezőgépben lévő film síkjának és az objektív térbeli irányának meghatározására is

179. Hogyan állítjuk elő a jelenet képét az ábrázolósíkban?

A jelenet képét tehát az (u, v) ábrázolósíkban definiált ablakra történő vetítéssel állítjuk elő. Ez egy középpontos vetítés, amellyel perspektivikus térhatást érhetünk el a keletkező képen.

180. Hogyan transzformáljuk a világkoordináta-rendszerben definiált objektumokat a normalizált látótérbe?

Ezzel azonban még mindig nem fejezhetjük be a kivágást és az ábrázolási térbe történő transzformálást, mivel különböző esetekben a látótestek mérete és formája még eltérő. (Ha ezen megfelelő egységesítéssel nem tudnánk változtatni, akkor a megjelenítési algoritmusoknak a legkülönbözőbb méretű látótestek kezelésére is fel kellene készülniük.) Ezért egy további affín koordináta-transzformációval a felhasználásspecifikus látóteret egy normalizált látótérbe transzformáljuk.

181. Milyen célt szolgál a w-clipping algoritmus?

A figyelembe veendő látótér meghatározása után, a kivágás utolsó lépéseként meg kell határoznunk, hogy a modelltér objektumai közül melyek esnek bele a látótérbe. Ehhez speciális kivágóalgoritmusokat használunk, amelyek közül legismertebb az ún. Cohen–Sutherland w-clipping.

182. Mi célt szolgálnak az outkódok a Cohen–Sutherland kivágó algoritmusban?

183. Mi a tárgypontosság-algoritmusok lényege?

A modelltér-eljárások vagy tárgypontosság-algoritmusok az objektumokat az adott nézőpontra vonatkoztatva közvetlenül hasonlítják össze. Ezek pontosságának csak a gépi számábrázolás szab határt.

184. Mi a képpontosság-algoritmusok lényege?

A képpontosság-eljárások a megjelenítendő kép pixeljei szerint határozzák meg a jelenetek látható részeit. Ezek pontossága nyilvánvalóan függ a megjelenítés eszközének felbontásától is.

185. Mit nevezünk back-face cullingnak?

Ha a poligonok normálvektorait úgy állítjuk be, hogy az objektumból kifelé mutassanak, akkor azok a poligonok, amelyek normálvektorai nem a néző felé mutatnak, biztosan takarva lesznek az objektum közelebbi felületei által. Ezeket a takarásba kerülő, úgymond hátrafelé néző poligonokat back-face poligonoknak nevezzük, és az objektumokat leíró adatbázisból való eltávolításukat eredményező eljárást pedig back-face cullingnak.

186. Mennyivel csökkenti a back-face culling a láthatósági algoritmusok műveletigényét?

Ha az ábrázolandó jelenet egyetlen, konvex poliéderről áll, akkor a látható felületek meghatározása egyetlen back-face culling műveletre egyszerűsödik. Más esetekben további vizsgálatok szükségesek, hiszen a front-face poligonok is takarhatják egymást.

187. Mire használhatók a befoglaló testek (bounding volume)?

A térbeli objektumok takarási viszonyainak meghatározását esetenként segítheti, ha ezeket egyszerűbb formájú testekbe zárjuk, amelyek térbeli elhelyezkedését hatékonyabb algoritmusokkal tudjuk tesztelni. Ezeket befoglaló testeknek (bounding volume) nevezik.

188. Mi a min/max teszt alapgondolata?

az objektumok vetületeit téglalapokkal vesszük körül, és ha ezeknek a téglalapoknak nincs közös része, akkor biztosak lehetünk abban, hogy a vetületeknek megfelelő felületek sem fedik egymást.

189. Mi a mélység-rendező algoritmus lényege?

hogy a megjelenítendő objektumokat a nézőponttól való távolság függvényében sorba kell állítani (depth-sort). Az algoritmus a helyes takarási viszonyokat úgy alakítja ki, hogy a nézőhöz közelebb eső objektum

képe felülírja a távolabbi objektum képét.

190. Milyen célra használható a BSP-fa algoritmus?

A háromdimenziós objektumok tetszőleges nézőponthoz tartozó megjelenítésére használt algoritmusok közül az egyik leghatékonyabb a bináris térfelosztó fákat (BSP-fák, BSP = Binary Space Partitioning) alkalmazó eljárás.

191. Mi a területfelosztó algoritmusok alapgondolata?

A területfelosztó algoritmusok alapgondolata, hogy néhány esetben (például a képen csak egy objektumot kell ábrázolni), nagyon egyszerűen megállapítható a képernyőn megjelenítendő kép, ezért a bonyolultabb takarási vizsgálatokat célszerű a kép területének kisebb részekre való rekurzív felosztásával visszavezetni az egyszerűen kezelhető esetekre.

192. Milyen puffereket használ a z-buffer algoritmus?

- a frame-buffert, amely a képernyő pixeljeihez rendelt színértékeket tárolja, induló feltöltése a képernyő háttérszíne,
- a z-buffert, amely az egyes pixelekhez rendelt z értékeket tárolja a normalizált látótérből, kezdeti értéke a hátsó kivágósík z koordinátája (a maximálisan ábrázolható z-érték).

193. Ismertesse a z-buffer algoritmus működését!

a z-buffert, amely az egyes pixelekhez rendelt z értékeket tárolja a normalizált látótérből, kezdeti értéke a hátsó kivágósík z koordinátája (a maximálisan ábrázolható z-érték).

194. Mennyiben függ a z-buffer algoritmus a modelltér objektumainak alakjától?

A z-buffer algoritmus a modelltér elemeinek formájától független, így háromszög, poliéder vagy görbült felületek láthatóságának megállapítására egyaránt alkalmas.

195. Mi a scan-line algoritmusok lényege?

A scan-line algoritmusok képpontosság műveleteket használva, pixelsorokként készítik a képet, és a poligonok (háromszögek) frame bufferbe történő soronkénti konvertálásának módszerén alapulnak.

196. Hogyan határozzák meg a kirajzolandó pixeleket a scan-line algoritmusok?

Ha a pixelsor egy szegmense csak egy háromszöghöz tartozik, akkor ezt egyszerűen csak meg kell jeleníteni. Ha egy pixel több szegmenshez tartozik, akkor a megfelelő háromszögek mélységi vizsgálatával kell eldönteni, hogy melyik háromszög felületi pontját kell kirajzolni.

197. Ismertesse a sugárkövetéses algoritmust!

A sugárkövetéses (raytracing) algoritmusok a felületek láthatóságát a nézőpontból kiinduló, képzeletbeli fénysugarak követésével határozzák meg. Tipikus képpontosság-algoritmusok.

198. Mi a szerepe a megvilágítási modelleknek a vektorgrafikában?

A megvilágítási modellekkel írjuk le a modelltér objektumainak és a fényforrásoknak a kapcsolatát.

199. Mitől függ a modelltér egy objektuma felületi pontjának színe?

- a fényforrás típusa (pontoszerű, kiterjedt stb.),
- a fényforrás sugárzásának erőssége és színe,
- a felület fényvisszaverő képessége,

- a felület és a fényforrás távolsága,
- a felület normálisa és a fénysugár által bezárt szög.

200. Milyen hatása van a renderelési időre, ha egy képelőállításnál több megvilágítási modellt is figyelembe veszünk?

Általában egy realisztikus fényviszonyokat tükröző kép megjelenítéséhez több modellt is felhasználunk, viszont minél többet veszünk figyelembe, a képelőállításnak annál nagyobb az erőforrásigénye (azaz annál lassúbb lesz a renderelés).

201. Hogyan kapjuk meg egy pixel végleges színét?

Egy felületi pontnak megfelelő pixel végleges színét úgy kapjuk meg, hogy a test eredeti alapszínéhez az egyes megvilágítási modellek alapján számított módosító világosság és színértékeket hozzáadjuk.

202. Sorolja fel a megvilágítási modelleket és alkalmazhatóságukat!

Az űrlap teteje

Megvilágítási modellek (2/4)

A fontosabb megvilágítási modellek a következők:

- általános háttérmegvilágítás (ambient light), amikor az objektumok saját színükkel egyenletesen sugároznak;
- diffúz fényvisszaverődés (diffuse reflection), amely a matt felületekre jellemző;
- tükröző fényvisszaverődés (total reflection), amely tükörként viselkedő felületekre jellemző;
- csillogó (fénylő) fényvisszaverődés (specular reflection), amely fénylő felületekre jellemző;
- átlátszóság (transparency), amely olyan testekre jellemző, amelyeken a fénysugár teljes egészében vagy részben áthalad;
- árnyék (shadow).

203. Mit nevezünk lokális megvilágítási modellnek?

A lokális megvilágítási modellekben az objektumok színét, világosságát más objektumok nem befolyásolják, ezek csak az objektumtól, a fényforrásoktól és a nézőponttól függnék. Ilyenek a háttérmegvilágítás, valamint a diffúz és csillogó fényvisszaverődés.

204. Mit nevezünk globális megvilágítási modellnek?

A globális megvilágítási modellekben az objektumok színe nemcsak közvetlenül a fényforrásoktól, hanem a más objektumokon áthaladó, illetve a más objektumok által visszavert fénytől is függ.

205. Mit jelent a háttérfény (ambient light)?

A háttérfény lokális megvilágítási modellben nincs fényforrás, az objektumok „saját” fényüket bocsátják ki. Ez megfelel annak, hogy a jelenetet egy irányfüggetlen, szórt fény (ambient light) világítja meg.

206. Mi jellemzi a diffúz fényvisszaverődést?

A diffúz visszaverődés (diffuse reflection) a matt, illetve durva felületek jellemzője, ezek a Lambert-féle megvilágítási törvényekkel modellezhetők. Ez esetben a visszavert fény sűrűsége független a kilépő sugár irányától, tehát a felület minden szögből egyformán fényesnek tűnik.

207. Mi az ideálisan tükröző visszaverődés?

Az ideálisan tükröző visszaverődést (total reflection) tapasztalhatjuk például egy síktükör esetében. Ebben az esetben a fénysugár a geometriai optikából jól ismert törvénynek megfelelően halad: a belépő és visszavert sugár, valamint a felületi normális egy síkban helyezkedik el, és a belépő és visszavert sugár a tükröző felület normálisával azonos szöget zár be

208. Mi az irányított diffúz visszaverődés?

Az olyan fényvisszaverődést, amelynek van egy kitüntetett iránya, amely irányban a felület a legtöbb fényt veri vissza, majd ahogy ettől az iránytól távolodunk, a visszavert fény mennyisége ennek megfelelően csökken, irányított diffúz visszaverődésnek (specular reflection) nevezzük.

209. Mikor kell alkalmazni a Phong-féle megvilágítási modellt?

Az irányított diffúz visszaverődés tapasztalati megfigyelési alapján javasolta Phong a róla elnevezett megvilágítási modellt tökéletlenül tükröző testek számára.

210. Miért egyszerűbb poliéderek megvilágítását kiszámolni?

Ha a test határolófelülete poligonokból áll, akkor (feltéve, hogy fényfolthatással nem kell számolnunk) elegendő a poligon egy pontjához tartozó szín és intenzitásértéket meghatározni és ezt követően a poligont ezzel a színnel feltölteni.

211. Mutassa be a lokális megvilágítási algoritmusok lényegét!

Ha a test határolófelülete poligonokból áll, akkor (feltéve, hogy fényfolthatással nem kell számolnunk) elegendő a poligon egy pontjához tartozó szín és intenzitásértéket meghatározni és ezt követően a poligont ezzel a színnel feltölteni. Így ebben az esetben a megvilágítás kiszámításának műveleti igénye nem a felületi pontok számával, hanem a testet borító poligonok darabszámával lesz arányos.

Erre a gondolatmenetre alapozva, a nem poligon határolta testeket a megvilágítás kiszámításához sokszögekkel, leggyakrabban háromszögekkel szokták közelíteni.

A sokszögekre bontott testekre dolgozták ki a lokális megvilágítási algoritmusokat, amelyekkel – ha a sokszögekre való felbontás elég finom – görbült felületek képét is valósághűen meg tudjuk jeleníteni.

212. Mit nevezünk shading-nek és melyek a legismertebbek?

Az objektumok felületeinek színét meghatározó eljárásokat árnyalási algoritmusoknak (shading) is szokták nevezni a szakirodalomban. A lokális megvilágítási modelleken alapuló árnyalási algoritmusok közül a legfontosabbak:

- a flat shading,
- a Gouraud shading,
- a Phong shading.

213. Mi a flat shading lényege?

Az objektumokat beborító poligonok árnyalásának a legegyszerűbb módja a constant vagy flat shading, amikor minden egyes poligont egy-egy azonos színnel jelenítünk meg.

214. Milyen megvilágítási modellt használunk a flat shading-nél?

Az eljárás a szórt háttérvilágítás és a diffúz visszaverődés együttes megvilágítási modelljét használja, ezért egy felületi sokszög színének meghatározásához elegendő a poligon normálvektorának kiszámítása (lásd a

megvilágítási egyenletet) és a fényforrás jellemzőinek figyelembevétele. Ez alapján számítjuk ki poligononként azt az egy értéket, amely meghatározza a sokszög színét.

215. Mi a Gourand shading lényege?

Realisztikusabb képek érdekében dolgozták ki az interpolált árnyalással működő algoritmusokat, amelyek a közelítő poligonokat (háromszögeket) nem azonos színnel árnyalják, hanem a sokszög pontjainak intenzitásértékét a csúcspontok intenzitásértékeiből lineáris interpolációval származtatják, ezáltal az éles intenzitáskülönbségeket elsimítják.

216. Ismertesse az intenzitás-interpoláló árnyalás lépéseit!

- Minden csúcspontban, ahol háromszögek találkoznak, kiszámítjuk a találkozó lapokhoz tartozó normálisok átlagát, így kapunk csúcspontonként egy ún. „pseudo-normális”-t (lásd ábra).
- A szórt háttérvilágítás és a diffúz visszaverődés megvilágítási modellje alapján kiszámítjuk a modelltérben a csúcsok intenzitásértékét a pseudo-normálisból.
- Vetítjük a háromszögeket a képsíkra és itt a csúcsok intenzitás értékeit lineárisan interpoláljuk az élekre és a lapokra.

217. Mi a lényege a Phong shading-nak?

A normálvektor-interpoláló árnyalás néven is ismert Phong-árnyalás a felületi normálvektorokat interpolálja, nem pedig az intenzitásértékeket. A normálvektorok interpolálása (hasonlóan a Gouraud árnyalás interpolálásához) a poligonok élein, majd a poligonok belsejében lévő pontokra történik meg. Végül minden pont számára az így kapott normálvektor felhasználásával számítják ki az intenzitásértéket a lokális megvilágítási egyenlet alapján.

218. Mi a legfontosabb különbség az előállított képen a Phong- és a Gouraud-árnyalás között?

Specular visszaverődési modell esetén jelentős különbség van a Gouraud- és a Phong-árnyalás között, az utóbbi sokkal élethűbben adja vissza a felületeken megjelenő fényfoltokat. Bár a Gouraud- és a Phong-modell közötti különbség leglátványosabb megnyilvánulása a fényfoltok megjelenítésében tapasztalható, általában is kedvezőbb eredményeket kapunk ez utóbbi algoritmus használatakor

219. Milyen megvilágítási modellel kezelhetjük a többszörösen visszaverődő és megtörő fényt?

Ezeket az effektusokat figyelembe vevő eljárásokat a szakirodalom globális megvilágítási modelleknek nevezi.

220. Mi a rekurzív sugárkövetés lényege?

a rekurzív sugárkövetéses algoritmus vagy raytracing több lépésben követi a visszaverődő vagy törő fénysugarakat és így képes a globális megvilágítási effektusok korrekt figyelembevételére.

221. Mi a radiosity és photon mapping eljárások közös jellemzője?

A fényterjedés fizikai törvényeit (fénysűrűség egyenlet, fotonok mozgása) „szimuláló” algoritmusok (radiosity, photon mapping) ezzel szemben teljesen szétválasztják a látható felületek meghatározását és azok árnyalását.

222. Melyik fotorealistikus képábrázoló algoritmus modellezi hatékonyabban a tükröződést, illetve a diffúz visszaverődést?

A nézőpontfüggő raytracing algoritmusok könnyedén kezelik a tükröződéseket, de egyes diffúz jelenségek kezelése (például kiterjedt világítótestek) nehézséget okozhat. Ezzel szemben a nézőpontfüggetlen

algoritmusok hatékonyan modellezik a diffúz jelenségeket, de a tükröződés figyelembevétele esetenként csak külön eljárásokkal oldható meg.

223. Melyek a raytracinggel készített képek fontosabb jellegzetességei?

- objektumok egymásra vetett árnyékai,
- többszörös tükröződések,
- átlátszóság kezelése fénytöréssel.

224. Mit értünk backward raytracing alatt?

a fény terjedésével ellentétben nem a fényforrásokból kiinduló sugarakat kezdi vizsgálni, hanem azokat a nézőpontból indított vetítősugarakat elemzi, amelyek visszafelé eljutnak a fényforrásba. Ezt a megoldást „hátrafelé történő sugárkövetésnek”, vagy backward raytracingnek nevezik.

225. Ismertesse a rekurzív sugárkövetés algoritmusát!

A rekurzív sugárkövetés algoritmusához a 3D-s térben definiáljuk a megfigyelő nézőpontját és egy ablakot, amelyet a képernyő felbontásának megfelelő számú cellára felosztunk. (Tehát például 800×600 -as felbontás esetén ez 480.000 cellát jelent.)

Az ablak minden egyes cellája tehát a képernyő, illetve a frame buffer egy pixeljének felel meg. A nézőpontból minden egyes cellán keresztül egy vetítősugarat indítunk, és meghatározzuk ezek metszéspontját a modellter objektumaival.

226. Mikor lesz egy pixelnek elsődleges színe a sugárkövetéses algoritmusnál?

Ha egy elsődleges sugár visszaverődve egy felület egy pontjáról „eltalál” egy fényforrást, és nem kell tükröződést illetve fénytörést figyelembe venni, akkor az elsődleges sugár képzésénél felhasznált cellának megfeleltetett pixel színe a megvilágítási modellből közvetlenül számítható.

227. Milyen célt szolgálnak az árnyéksugarak?

Ha a fényforrással összekötő sugarak (ezekből annyit kell indítani, ahány fényforrás van definiálva a modellterben), amelyeket árnyéksugaraknak (shadows ray) is neveznek, beleütköznek valamilyen objektumba mielőtt a fényforrást elérnék, akkor a felületi pont árnyékban van, és a megfelelő fényforrást ki kell hagyni a felületi pontot árnyaló megvilágítási egyenletről.

228. Miként alkalmazzuk a befoglaló testeket a raytracing során?

Ennek során legtöbbször gömböket használunk fel. Ez azt jelenti, hogy olyan legkisebb méretű gömböket definiálunk a modellterben, amelyek egyes objektumainkat teljes mértékben tartalmazzák.

229. Jellemezze a radiosity algoritmus felhasználhatóságát és fizikai elvét!

A radiosity algoritmus alkalmazása esetén először a jelenet összes objektumának felületére meghatározzák a visszavert és a felület saját sugárzásából adódó fénymennyiséget, amelyből származtatható a globális megvilágítási modell. Ennek erőforrásigénye, és így a gépidőszükséglete általában a többi renderelési algoritmushoz képest többszörös. A fényviszonyokat ugyanakkor csak egy alkalommal kell kiszámítani. Így például egy nézőpont (kameraállás) megváltoztatása esetén – feltéve, hogy a jelenethez tartozó fényforrások és objektumok helyüket nem változtatják – az új kép a megismert egyszerű láthatósági algoritmusokkal már nagyon gyorsan kiszámítható a változatlan megvilágítási modell miatt. (Ez lényegesen gyorsabb, mint egy raytracelt kép előállítás.)

230. Mikor lassú és mikor gyors a radiosity eljárás?

231. Hogyan kezeli a fényforrásokat a radiosity eljárás?

A radiosity eljárásnál viszont minden felületet fénysugárzónak tekintünk, és a fényforrások sem pontszerűek, hanem véges kiterjedésű felületekkel lesznek modellezve.

232. Mit lehet modellezni a photon mapping eljárással?

Ezzel a globális megvilágítási eljárással fotorealistikusan modellezhetők a közvetett megvilágítás hatásai, valamint az úgynevezett kausztikus optikai jelenségek.

233. Foglalja össze a photon mapping algoritmus lényegét!

A fotonok útját a fényforrásból kiinduló nyomkövetéssel (path tracing) kísérik figyelemmel, és adataikat ún. foton térkép (photon map) segítségével tartjuk nyilván.

234. Mit tartalmaz a fotontérkép?

A fotonok útját a fényforrásból kiinduló nyomkövetéssel (path tracing) kísérik figyelemmel, és adataikat ún. foton térkép (photon map) segítségével tartjuk nyilván.

235. Hogy számítjuk ki egy pont fényintenzitását a photon mapping alkalmazása esetén?

a következő komponensek összegeként határozzuk meg:

közvetlen megvilágítás valamelyik fényforrástól;
tükröző visszaverődés;
kausztikus jelenségek;
közvetett megvilágítás.

236. Miért használhatunk fel látható felület meghatározó algoritmusokat az árnyékok meghatározására?

A láthatósági algoritmusok azt határozzák meg, hogy mely felületek láthatók a nézőpontból, az árnyékokat előállító algoritmusok pedig azt, hogy mely felületek „láthatók” a fényforrásból. Így a látható felület meghatározó és az árnyékképző algoritmusokat lényegében azonosnak tekinthetjük.

237. Sorolja fel az árnyékképzéshez felhasználható egyszerűbb algoritmusokat!

- scan-line;
- árnyéktetek használata;
- z-buffer.

238. Hogyan határozhatjuk meg az árnyékokat z-buffer algoritmussal?

Az algoritmus a fényforrást használva vetítési középpontként, kitölt egy z-bufferet, amelyben a fényforrástól mért távolságot rögzítik. Ezután a korábban megismert z-buffer algoritmus meghatározza a látható felületi pontokat, de az egyes pontok árnyalásakor a pontok térbeli koordinátáit átranzformálja a fényforrás középpontú rendszerbe, és a z koordinátákat összeveti az árnyék z-bufferben lévő távolságértékekkel. Ha a tranzformált pont z koordinátája nagyobb, mint a bufferben tárolt érték, akkor a vizsgált pont távolabb van a fényforrástól, tehát árnyékban van.

239. Hogyan használhatjuk a felületi részlet poligonokat?

Ezeket a poligonokat egyszerűen felvisszük a bázisfelület megfelelő oldalára, és az árnyalás során ezek a

poligonok egyszerűen helyettesítik a bázisfelület megfelelő részét az általuk fedett területeken.

240. Mit nevezünk textúrának és texel-nek?

Kétdimenziós képek felületekre történő leképezésével a Catmull által kidolgozott textúrázás (texture mapping) néven ismert technika alternatív lehetőséget biztosít objektumaink megjelenítésének javítására. A művelet során felhasznált képet textúrának (texture map), egyes elemeit pedig texelnek hívjuk.

241. Hogyan tudjuk figyelembe venni a textúrákat a képernyő-koordinátarendszerben?

Textúrák alkalmazása esetén a felületnek megfelelő képernyőrész minden egyes pixelje esetén meg kell vizsgálni, hogy a hozzátartozó színértéket a textúrák megfelelő részei mennyiben módosítják. Ehhez többlépcsős leképezéssel egyértelmű kapcsolatot kell létesítenünk a textúra u , v koordinátarendszere és a képernyőkoordináta-rendszer között

242. Mit nevezünk algoritmikus textúrának?

Algoritmikus textúrák esetén a felületre feszített mintázatot programmal állítjuk elő. Erre jó példa egy fraktálok segítségével generált fűszálakból álló rét.

243. Mit nevezünk edge-antialiasingnek?

A textúra elemekből képzett ferde vonalak lépcsőzetes kialakulásának megakadályozását célozza az edge anti-aliasing. Ennek lényege, hogy a vonal széleihez tartozó texelek színét átlagoljuk a vonal melletti texelek színével, és ezt az átlagértéket jelenítjük meg a képen. Ezzel a „simító” eljárással a kapott átmeneti színek elmosóak a képen a „lépcsőket”.

244. Mi a hatása a bilinear filteringnek?

A textúra egy képpontjának színét a függőleges és vízszintes irányban mellette lévő 4 pixel színértékének átlagából számítjuk ki. Hatására a textúra elemek közötti éles átmenetek elmosódnak

245. Mit jelent a mip-mapping?

Ennél az eljárásnál egy textúrát több különböző felbontásban is letárolunk. Ha közeledünk a nézőponttal egy textúrázott objektumhoz, akkor a pixelesedés kizárása érdekében egyre finomabb felbontású változatát feszítjük fel a textúrának az objektum felszínére. Minél közelebb van egy objektum, annál részletesebb a textúra, a távolabbi tárgyak pedig elmosódottabbak (lásd a mellékelt ábrát).

246. Hogyan lehet perspektivikus hatást elérni textúrákkal?

A térhatás elérése érdekében a textúrákat a távolság függvényében kisebbítve, ferdítve visszük fel a felületre (lásd a mellékelt ábrát).

247. Hogyan modellezhetjük az átlátszó és áttetsző testeket?

- törésmentes átlátszósági eljárások, amelyek lehetnek interpoláló, illetve szűrő algoritmusok,
- törő átlátszósági eljárások, amelyek a fénytörés törvényei szerint kezelik a testeken áthaladó fénysugarakat.

248. Mire és hogyan használható az alpha-blending?

Ehhez az algoritmuscsoporthoz tartozik az átlátszóság textúrákkal való kezelését megvalósító alpha blending. Erre legjobb példa az áttetsző víz ábrázolása, amikor a víz felszínének textúráját részlegesen „átlátszóvá tesszük”, és „mögötte” megjelenítjük a vízfenék textúráját.

Ehhez legtöbbször texelenként az RGB színértékek mellett az áttetszőség mértékét meghatározó α csatorna

értéket (RGBA színkódolás) használjuk fel.

249. Milyen ismert algoritmus kezeli a törő átlátszóságot?

a fotorealistikus képmegjelenítés raytracing algoritmusával valósíthatjuk meg.

250. Mit nevezünk objektumok közötti tükröződésnek?

ha egy felületen a jelenet egy másik elemének képe visszatükröződik. Ez a jelenség lehet nézőpontfüggő, tükröszerű visszaverődés, de lehet a nézőponttól független diffúz visszaverődés is.

251. Mi az environment mapping lényege?

Az environment mapping eljárással textúrákkal modellezhetjük egy objektum környezetében lévő tárgyak visszatükröződését az objektum felszínén.

252. Mit nevezünk reflection mapping-nek?

Ennél a megjelenítendő tükröző objektum köré egy gömböt definiálnak, amelyre rávetítik az objektumot körülvevő modelltér képét. A gömb felületét ezután 2 dimenziós textúraként használják fel a tükröző felület képének kialakítása során.

254. Hogyan modellezhető a köd?

Ködöt fotorealistikus képeken úgy tudunk ábrázolni, hogy a ködnek megfelelő szint (általában fehér vagy fehéresszürke) a nézőponttól távolodva egyre nagyobb arányban keverjük a képhez.

255. Hogyan lehet modellezni a füstöt és a lángot?

Füstöt általában speciális textúrákkal modellezhetünk. A füstnek megfelelően mozgatjuk az ábrázoló textúraelemeket, melyek részben átlátszóak, hatásuk elmosódott.

Hasonló módszerrel tudunk előállítani például lángot is.

5. A SZÁMÍTÓGÉPES GRAFIKA SZABVÁNYAI

256. Mi okozta, hogy a hardver hordozhatósága ma már általánosan elfogadott követelmény a számítógépes grafikában?

Kezdetben a grafikus szoftvereket eszközfüggően kellett programozni, ez azonban egyre kevésbé felelt meg az igényeknek. Egyrészt a felhasználók már nem fogadták el függő helyzetüket az eszközgyártóktól, másrészt a grafikus szoftvereket fejlesztő szoftverházaknak is elemi érdekévé vált, hogy az egyre bonyolultabb és növekvő fejlesztési költségekkel előállított programcsomagokat minél több hardverplatformon értékesíteni tudják.

257. Sorolja fel, hogy a szabványosítás milyen területekre terjed ki a számítógépes grafikában!

- grafikus felhasználói felület szabványai,
- grafikus programcsomag szabványok,
- a felhasználói programok és a grafikus programcsomagok közötti interfészt, az API-t (Application Program Interface) meghatározó szabványok,
- a grafikus rendszerek közötti adatcserét biztosító fájl szabványok,
- a színkezelés, a betűformák szabványai,
- a különböző hardver csatolókra (BIOS, driverek) és grafikus eszközökre vonatkozó szabványok.

258. Mi az X-Windows System?

Az X-Windows System hardverfüggetlen, a raszteres grafikára alapozott ablakozó megjelenítés szabványa.

259. Az X11 szabvány szerint mit tartalmaz a grafikus munkahely?

A grafikus munkahely az X11 szerint képernyőt (screen), billentyűzetet (keyboard) és mutatóeszközt (egér, tablet) kezelő architektúra, amelyen az X-szerver program fut.

260. Mi az X-szerver feladata?

a grafikus megjelenítés (2D-s rajzolás) és a felhasználó inputjainak a fogadása. Az X-szerver hálózati kliensekkel, az úgynevezett alkalmazásokkal (applications) tarthat kapcsolatot, amelyek számára hozzáférést biztosít a grafikus munkahelyhez.

261. Mi a feladata a Windows Manager kliensnek?

A X11 felhasználói felülete a Windows-ból jól ismert ablakozó rendszer. Az ablakok méretezését és általában a menedzselését a Windows Manager program végzi, mely az X-szerver számára szintén egy kliens.

262. Mi az SRGP?

Ezt a szabványt egyszerű rasztergrafikus programcsomagnak, vagy az angol megnevezése után rövidítve SRGP-nak (Simple Raster Graphics Package) nevezik.

Az SRGP tulajdonképpen a legalapvetőbb rasztergrafikus funkciókat megvalósító programegységekre vonatkozó szabvány, amely X11 alatt alkalmazható.

263. Milyen követelményeket határoztak meg a GKS szabvánnyal szemben?

- egységes illesztési helyet meghatározni a grafikus rendszerek és az egyedi alkalmazások között,
- a felhasználástól független, számítástechnikailag hatékonyan realizálható könyvtár definiálása a vektorgrafikus rendszerek fontosabb funkcióira,
- a grafika területén minél több általánosítható követelményt lefedni a szabvánnyal,
- elválasztani a grafikus rendszerek alap- (ún. mag) funkcióit a magasabb szintű modellezési funkcióktól,
- a szabvánnyal egy fejlesztési irányt mutatni a készülékgyártók számára.

264. Mi a PHIGS?

A PHIGS a szabvány angol elnevezésének (Programmers Hierarchical Interactive Graphics System) rövidítéséből származik. Ezek szerint a PHIGS a programozók hierarchikus, interaktív grafikus rendszere.

265. Mi az OpenGL?

Az 1990-es években, különösen a 3D-s gyorsító chipek fejlődésével egyre nagyobb gondot okozott, hogy a hardverben megvalósított láthatósági, megvilágítási és árnyalási algoritmusokat a szabványok nem tartalmazták. A Silicon Graphics fejlesztői ezekre a problémákra kerestek megoldást, amikor a korábbi IRIS GL grafikus szoftverük tapasztalatainak felhasználásával egy szoftverinterfészt specifikáltak OpenGL néven a grafikus kártyák hardveréhez.

266. Milyen „magasabb szintű”, új renderelő funkciókat tartalmaz az OpenGL a korábbi szabványokhoz képest?

- texture mapping, mip mapping,

- alpha blending (átlátszóság),
- légköri effektek (köd, pára, füst),
- anti-aliasing.

267. Hogyan adjuk meg a primitíveket az OpenGL-ben?

A grafikus információkat a rendszerrel csúcsponti adatok (vertex) formájában közölhetjük.

268. Mikor történik a fényforrások hatásának a figyelembevétele az OpenGL-ben?

A következő lépésben az OpenGL a primitíveket egy vetítéssel leképezi a látótérbe. Ebben a fázisban történik a fényforrások kihatásainak figyelembe vételéhez szükséges számítások elvégzése is.

269. Mi történik a raszterizálás során az OpenGL-ben?

Az úgynevezett raszterizálással megtörténik az átalakítás 2D-s egész koordinátákra. Ekkor veszi a rendszer figyelembe a különböző bitmap és pixelmezőket, amelyek közé számítják a textúrákat is. A raszterizálás során kapott pontokhoz a színértékek és a pont z értéke is hozzá lesz rendelve.

270. Mire szolgálnak a képtöredékek (fragments) az OpenGL-ben?

A raszterizálás után kapjuk a képtöredékeket (fragments), amelyek alapján z-buffer eljárással határozzák meg a végső láthatóságot. Az OpenGL (lásd az előző lap ábráján a szaggatott vonallal jelöltek) megengedi, hogy a frame bufferből adatokat kiolvassunk, illetve egyes tartományokat kimásoljunk.

271. Milyen a feladatmegosztás az OpenGL és a Windows között?

A rendering pipeline utolsó lépését a frame bufferből való képkirajzolást a monitor képernyőjére nem az OpenGL, hanem az alkalmazott ablakozó rendszer hajtja végre. Ez összhangban van az OpenGL alapkoncepciójával, amely szerint a konfiguráció és grafikus input készülék vezérlése a Windows feladata. Így például a Windows kezeli a képernyő ablakait, a szintáblázatot stb.

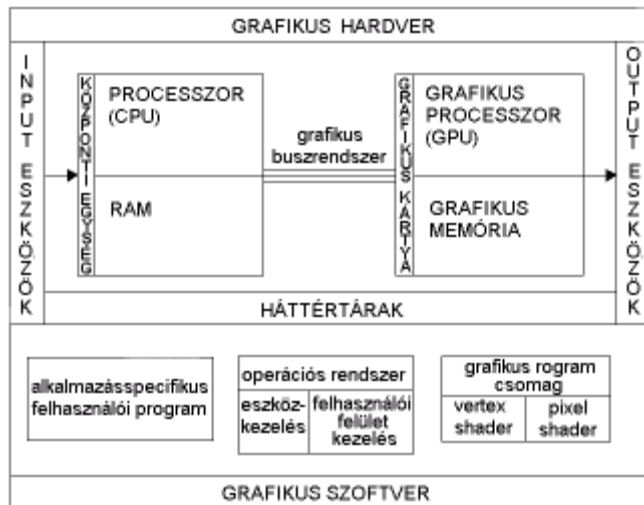
272. Jellemezze a Direct 3D „házi” szabványt!

A Direct 3D API funkcionálisan rendkívül hasonlít az OpenGL-re. Így például a 3D-s objektumokat poligonfelületekkel modellezi, azonosak a modelltér és megjelenítés transzformációs lehetőségei, hasonló a láthatósági, valamint a megvilágítási és árnyalási algoritmusok kezelése, a speciális effektek (pl. köd) biztosítása.

A Direct 3D jó animációs lehetőségeket biztosít, erre is visszavezethető, hogy főként a PC-s játékprogramoknál használják elterjedten.

6. A GRAFIKUS HARDVER ÉS SZOFTVER

273. Milyen hardver- és szoftver-elemek alkotják a grafikus rendszer architektúráját?



A grafikus rendszer architektúrája

274. Sorolja fel a raszteres input-eszközöket és ezek feladatait!

A raszteres input eszközök – szkennerek, digitális kamera és fényképezőgép (lásd az ábrán) – segítségével fekete-fehér vagy színes képeket, filmeket vihetünk be képpontonként digitális formában a grafikus rendszerbe.

275. Milyen vektoros input-eszközt ismer?

Vektoros input eszköz a digitalizáló tábla (tablet), amely koordinátaadatok bevitelére szolgál, a szálereszt helyzetét a grafikus rendszer vektorkoordinátaként értelmezi

276. Milyen grafikus output-eszközöket ismer?

Raszteres output eszköz a monitor. Biztosítja a grafikus program és a felhasználó kommunikációját, így megjeleníti a modelltér objektumait munka közben képernyőn (pl. huzalvázás forma), és lehetővé teszi a munka végtermékének (pl. kép, film) megtekintését.

277. Mire szolgálnak a háttértárak a grafikus rendszerben?

A háttértárak (mágneselem, pendrive, CD, DVD) a grafikus szoftver és a hozzátartozó adatállományok (pl. textúrák), a vektorgrafikus adatbázis, valamint a grafikus rendszerrel előállított raszteres eredményadatok (film, képek) tárolására szolgálnak.

278. Mi a központi processzor szerepe egy vektorgrafikus rendszerben?

A központi memóriába (RAM = Random Access Memory) töltődnek be és a processzoron (CPU = Central Processing Unit) futnak a grafikus szoftverek (alkalmazásspecifikus felhasználói program és grafikus programcsomag). Ennek megfelelően a processzor végzi a modelltérben végrehajtott műveleteket, a jelenet objektumainak felbontását háromszögekre, és a jelenet háromszögeinek adatait továbbítja a grafikus kártya processzorához.

279. Mi a feladata a grafikus kártyán található processzornak, és milyen részekből áll?

a kártyán található grafikus processzor (2D, 3D gyorsító csip) a processzortól kapott háromszögszámok alapján elvégzi a megjelenítéshez szükséges geometriai transzformációkat és a raszteres kép előállítását is a képmemóriában (frame buffer). Az előbbi feladatot végrehajtó processzorrészt geometriai vagy vertex processzornak, az utóbbi, vektorraszter konverziót végző áramköröket raszterprocesszornak is szokták nevezni.

280. Milyen szoftverelemekből épül fel egy grafikus rendszer szoftver-architektúrája?

Az egyes szoftverelemek fontosabb feladatai a következők:

- Operációs rendszer
 - az ablakozó rendszer kezelése és a frame-bufferből való végső kirajzolás a képernyőre,
 - grafikus I/O eszközkezelés.
- Alkalmazásspecifikus felhasználói program
 - interaktív kapcsolattartás a felhasználóval,
 - alkalmazási területtől függő funkciók,
 - 3D modellezés,
 - objektummozgatás, animáció,
 - objektumstruktúrák kezelése (jelenetgráfok = scene graph),
 - grafikus adatbázis-kezelés,
 - jelenet előkészítése rendereléshez (fényforrás, kameradefiníciók stb.)

281. Milyen együttműködés van a felhasználói program, a grafikus programcsomag és a grafikus processzor programja között?

A felhasználói program a rendereléshez a grafikus programcsomagnak általában háromszögekből álló primitívek formájában adja át az adatokat vagy vertexek (kontrollpontok) megadásával „megrendeli” a felület ill. görbemodellezést, a grafikus programcsomag pedig a GPU-nak a háromszögekre vonatkozó vertexek koordinátáit, szín- és textúraadatait adja tovább.

282. Mi a feladata a modellező szoftvernek?

a modellező szoftverek feladata a felhasználóval való interaktív kapcsolattartással a 3D modellter objektumainak létrehozása, képernyőn való bemutatása, mozgatása, komplex objektumok és jelenetek összeállítása.

283. Mi a feladata a renderelő szoftvernek?

A renderelő szoftverek feladata a modellter jeleneteiből a raszteres kép előállítása (képszintézis).

284. Melyek a rendering pipeline fázisai, és milyen feladatokat látnak el?

- Modellezési fázis: e lépésben megtörténik a felhasználói program objektumainak átalakítása a renderelő szoftver adatstruktúrájára (pl. felbontás háromszögekre), illetve ez az adatstruktúra aktualizálásra kerül.
- Geometriai fázis: ekkor valósul meg a jelenet objektumainak (háromszögek) transzformálása (kamera, projektív transzformációk, kivágás stb.) és a megvilágítás figyelembevétele.
- Raszterizációs fázis: e munkafázisban a geometriai primitívekből kiszámításra kerülnek a textúrák figyelembe vételével a pixelek színekódjai.

285. Miért fejlesztették ki az MMX processzorokat?

kifejlesztésének célja az volt, hogy a hardver képes legyen a multimédia egyre inkább növekvő igényeit kiszolgálni.

286. Milyen indokai voltak az SSE utasításkészlet kifejlesztésének?

287. Milyen előnnyel jár a többszálás és többmagos processzorok alkalmazása a számítógépes grafikában?

A vektorgrafika modellterében végrehajtott és a rendering pipeline-nak a modellezés szintjén működő algoritmusai legtöbbször jól párhuzamosíthatók. (Gondoljunk pl. a raytracing esetében arra, hogy az egyes sugarak követése lépésenként egymástól függetlenül végrehajtható.) Ezért a processzorgyártásban 2004-ben, ill. 2005-ben bevezetett többszálás (hyper-threading) és kétmagos (dual core) technológia (Intel Prescott, ill. Smithfield) jelentős teljesítménynövekedést eredményezett a számítógépes grafikában is

288. Milyen részegységekből épül fel a monitorvezérlő kártya?

- Grafikus síninterfész: puffereken keresztül biztosítja a kártya kapcsolatát az AGP vagy a PCI Express sínnel;
- Videomemória (VRAM): DRAM memória a megjelenítendő digitális kép (frame buffer) és a 2D és 3D gyorsító algoritmusokhoz szükséges adatállományok (pl. textúrák) tárolására;
- Grafikus processzor: vezérli a frame buffer feltöltését, és hardverúton megvalósítja a rendering pipeline geometriai és raszterizációs grafikus algoritmusait (2D és 3D gyorsítás);
- Video-ROM: a kártyához tartozó video ROM-BIOS-t, valamint a grafikus módok és a karakterkészletek definícióit tartalmazza;
- RAMDAC (Random Access Memory Digital-to-Analog Converter): a képernyőn történő megjelenítéshez szükséges digitális-analóg konverziót végzi, előállítva az analóg RGB jeleket a monitor számára. (A grafikus kártya digitális-analóg konvertere, a RAMDAC teljesítményének és a monitor képfrissítési frekvenciájának és felbontásának is meg kell felelnie egymásnak.)

289. Miben mérjük a monitorvezérlő kártyák teljesítményét?

A grafikus kártyák teljesítményét frame/sec (FPS)-ben mérjük, amely a másodpercenkénti teljes raszteres képernyő megjelenítések számát jelenti.

290. Hogyan számítjuk ki a frame-buffer memóriaigényét?

A memóriaszükségletet bitben úgy kaphatjuk meg, ha a pixelszámot összeszorozzuk a színkód bitszámával.

291. Mi a 2D/3D-s gyorsító chipek alkalmazásának lényege?

A 2D/3D gyorsítás lényege az, hogy a két-, illetve a háromdimenziós képek előállításához szükséges számítások egy részét a grafikus kártyán lévő csip (GPU) veszi át, ezáltal tehermentesíti a központi processzort (CPU).

292. Mi a lényege a stream modellnek?

A grafikus processzor felépítésére a stream modellt alkalmazzák. Ennek két lényeges eleme van: az adatfolyam vagy stream és az ezt feldolgozó műveletvégző egység, a kernel.



293. Mi a feladata a geometriai processzornak?

a geometriai processzor végzi el vertex feldolgozó egységeivel a regiszterekben átadott háromszög vertex adatfolyamon először a transzformációs és megvilágítási számításokat, majd ezt követően végrehajtja a kivágást és a képernyőablakra történő vetítést.

294. Mi a feladata a raszterprocesszornak?

a raszterprocesszor a háromszögek csúcsponti értékei alapján interpolációval kiszámítja a belső háromszögpontok szín és megvilágítási értékeit, az így kapott pixelek adatait továbbadja a pixelfeldolgozó egységeknek. Itt megtörténik a végső láthatóság kiszámítása (z-buffer), a textúrázás és a speciális effektusok hatásainak figyelembevétele.

295. Mi a grafikus sínrendszer feladata?

A grafikus sínrendszer feladata a központi egység (CPU és RAM) és a GPU közötti adatátvitel biztosítása

296. Milyen előnnyel jár az OpenGL felhasználása egy grafikus alkalmazás fejlesztésében?

Ha magas szintű C programnyelven kell kifejleszteni egy grafikus rendszert (pl. geometriai modellezőszoftver, CAD rendszer vagy játékprogram), akkor az ehhez szükséges grafikus programcsomag funkciókat legcélszerűbben az OpenGL API grafikus parancsainak alkalmazásával lehet felhívni a felhasználói programban. Ennek előnye többek között az is, hogy az OpenGL-t használó alkalmazások hordozhatók, több géptípuson és operációs rendszer alatt is futtathatók.

297. Hogyan használjuk egy C programban az OpenGL függvényeket és állapotváltozókat?

az egyes OpenGL funkciókat a megfelelő OpenGL függvény meghívásával aktivizálhatjuk. A függvényhívó parancs felépítése:
`glFuggvenynev()`

A rendszerben azokat a globális paramétereket nevezzük állapotváltozónak, amelynek értékét figyelembe véve hajtja végre az OpenGL az egyes parancsokat. Ilyen pl. a grafikus objektum színe. Ezt nem kell minden parancsban beállítani, mert a rendszer az állapotváltozót addig használja, amíg azt a programozó meg nem változtatja.

298. Sorolja fel az OpenGL fontosabb alapfunkcióit!

- Geometriai modellezési parancsok: OpenGL primitívek, gömb, kúp, tórusz, Bézier és NURBS görbék és felületek.
- Szín, árnyalás és megvilágítási parancsok: RGBA színek, Gouroud és flat shading, fényforrások, diffúz, tükröző visszaverődés, csillogás.
- Textúrakezelési parancsok: textúradefiniálás, mip mapping, szűrők.
- Speciális effektusok parancsai: átlátszóság, anti-aliasing, kód.
- Raszteres objektumokat kezelő parancsok: bittérképek és karakterek, pixeles képek (nagyítás,

kicsinyítés).

- Pufferkezelő parancsok:

színpuffer (itt keletkezik a kép),

z-puffer (a z-buffer algoritmus z távolságértékének kezelésére),

gyűjtő puffer (több kép összegzésére használható).

299. Milyen lehetőséget biztosítanak a fejlesztőnek a programozható geometriai és raszterprocesszorok?

ua.

300. Milyen feladatot lát el a vertex shader program?

A vertex shader program minden egyes vertex esetében megkapja a helykoordinátákat (beleértve a normálvektort is), a textúra-koordinátákat és színvektort, az objektumokra (háromszögekre) vonatkozó állapotadatokat (pl. transzformációs mátrixok, anyagjellemzők stb.). Ezek alapján elvégzi a transzformációs és megvilágítási számításokat, és átadja a megváltoztatott hely, textúra, színkoordinátákat

301. Mi a feladata a pixel shader programnak?

A pixel shader program (OpenGL terminológiában fragment shader) minden egyes pixelre (fragmentre) megkapja a helykoordinátákat, a szín- és textúrakoordinátákat, elvégzi a végleges szín- és textúraszámításokat és átadja a számított színvektort és „z” értéket

302. Sorolja fel a számítógépes tervezés néhány részterületét, és adjon meg néhány professzionális tervezőprogramot!

- A számítógéppel segített tervezés (CAD), amelynek legfontosabb részterületei a géptervezés és az építészet. Ilyen szolgáltatásokat nyújtó programcsomagok például az AutoCAD, a CADKEY és az ArchieCAD.

- A különböző grafikák, például reklám célra történő számítógéppel történő előállítás. Ezek lehetnek különböző termékismertetési vagy oktatási célú prezentációk, de lehetnek képzőművészek által készített képek is. A vektoros rajzolóprogramok közül a legnagyobb múlttal rendelkezik és a legismertebb a CorelDRAW.

- A nyomdai kiadványszerkesztés (DTP) ma már elképzelhetetlen a számítógépes grafika felhasználása nélkül. E területen jól ismert és alkalmazott programok az Adobe Photoshop és az Adobe PageMaker.

- A szórakoztató és a filmipar is egyre inkább hasznosítja a számítógépes grafikát (lásd az ábrát). Ezekkel például elképzelt, sci-fi világokat lehet alkotni vagy 3D-s virtuális "szörnyeket" lehet készíteni. E területen ismert programok közül a 3D Studio MAX, a SOFTIMAGE 3D és a Maya említhető meg.

303. Milyen célokra állíthatunk elő grafikákat például a CorelDRAW-al?

A különböző grafikák, például reklám célra történő számítógéppel történő előállítás. Ezek lehetnek különböző termékismertetési vagy oktatási célú prezentációk, de lehetnek képzőművészek által készített képek is.

304. Milyen programcsomagokat használnak fel a DTP területén?

A nyomdai kiadványszerkesztés (DTP) ma már elképzelhetetlen a számítógépes grafika felhasználása nélkül. E területen jól ismert és alkalmazott programok az Adobe Photoshop és az

Adobe PageMaker.

305. Nevezzen meg legalább három programcsomagot, mellyel a filmipar számára készítenek animációkat!

A szórakoztató és a filmipar is egyre inkább hasznosítja a számítógépes grafikát (lásd az ábrát). Ezekkel például elképzelt, sci-fi világokat lehet alkotni vagy 3D-s virtuális "szörnyeket" lehet készíteni. E területen ismert programok közül a 3D Studio MAX, a SOFTIMAGE 3D és a Maya említhető meg.

306. Mire használható leginkább a 3D Studio Max?

Az Autodesk cég által kínált 3D Studio Max (rövidebben 3ds Max) egy vektorgrafikus modellező és animációs programcsomag, amelyet leginkább játékfejlesztéshez, ipari és építészeti tervek vizualizációjához, mozifilmek vizuális effektusaihoz és oktatófilmek készítéséhez használhatunk.

307. Sorolja fel a 3D Studio Max legfontosabb lehetőségeit!

- Valóságghű karakterek létrehozására alkalmas karakterfejlesztési funkciókkal rendelkezik, beleértve a ruha- és fejszimulációs eszközöket is. Támogatja a karakteranimációt.
- Több modellező eszközt kínál, mint például a primitívekkel való modellezés (CSG műveletek lehetőségével), kis- és nagyfelbontású poligonmodellezés, spline-modellezés (NURBS). Poligonmodellezésnél az ún. NURMS (Non-Uniform Rational MeshSmooth) technikának köszönhetően, a poligonháló éleinek geometriáját finoman lehet változtatni.
- Lehetővé teszi egy részecske-kibocsátó objektum létrehozását, amelyet havazás, felhő vagy spray modellezésére használhatunk.
- A felületek reális anyagi mintáknak megfelelő textúrázására (fa, márvány stb.) lehetőség van, biztosítva az UV koordinátákban megadott textúrák pontos illesztését az objektumok geometriájához.
- A programcsomag számtalan fényeffektust képes kezelni, mint például lencsecsillogás, füst, csillagköd, tűz stb.

308. Melyik tervezőprogram a legelterjedtebb a PC alapú rendszerek körében, és melyek a fő felhasználási területei?

Az Autodesk cég által kifejlesztett AutoCAD áll az első helyen a CAD tervező programcsomagok piacán.

309. Milyen szakmaspecifikus szoftverek egészítik ki az AutoCAD alap-programot?

310. Mire használható a VIZ Render?

A műszaki látványtervek előállításához is léteznek olyan programcsomagok, amelyek szervesen illeszkednek az AutoCAD-hoz. Ilyen például a VIZ Render, amely a 3ds Max-technológiára alapozva, lehetővé teszi az AutoCAD-el elkészített tervek (DWG fájlok) gyors fotorealistikus képi megjelenítését.

311. Milyen fontosabb részekből áll a Corel programcsomag?

CorelDRAW – vektoros rajzoló és oldaltervező programot,
Corel PHOTO-PAINT – raszteres fényképszerkesztő programot,
Corel TRACE – raszteres képet vektorossá átalakító programot (lásd az ábrát),

Corel R.A.V.E. (Real Animated Vector Effects) – Internetes vektoralapú animációkészítő,
Corel CAPTURE – a képernyő képeinek „elkapására” szolgáló szoftvert.

312. Mit tartalmaznak a Corel objektumkönyvtárak, és miért előnyös a használatuk?
több 10000 clipartot és digitális képet, valamint kb. 1000 OpenType fontot tartalmaz.

313. Mire alkalmas az Adobe Photoshop programcsomag?
alkalmas nyomdába vagy Internetre szánt képek előkészítésére, digitális felvételek retusálására, vagy akár montázsgrafikák elkészítésére.

314. Mutassa be az Adobe Photoshop nyomdatechnikai lehetőségeit?
A nyomdatechnikában a színes képeket a CMYK színtér komponensei szerint kell felbontanunk (color separation). Ehhez a Photoshop lehetőséget biztosít olyan színkivonat (color plate) képek előállítására, amelyek a CMYK színtér alapszíneinek megfelelő színek árnyalatait adja vissza. Ezeket a nyomda az ún. színnyomatok készítésénél használhatja fel.

315. Mondjon példákat az Adobe Photoshop filtereire!
A képek manipulálását és speciális effektusok előállítását az ún. filterek (szűrők) segítik. Így például a blur filter lágyítja a színek közti átmenetet. Ennek mértékét beállíthatjuk és akár koncentrikus körök mentén is alkalmazhatjuk. Utóbbi esetben például a Spin (perdület) paranccsal a képen a forgó mozgást szimulálhatjuk. A distort filterrel eltorzíthatjuk a képet például egy másik kép adatai alapján, a ripple filterrel pedig vízben tükröződő képet állíthatunk elő.

316. Mire használható a Maya programcsomag?
Professzionális funkciókat biztosít többek között a karakteranimáció, a vizuális effektusok és a trükkök területén.

317. Ismertesse a Maya által nyújtott fontosabb lehetőségeket!

- fejlett NURBS és polygon-modellezési, valamint textúrázási eszközök;
- hierarchikus felületfelosztásos modellezés (Subdivision Surface Modeling);
- rugalmas karakterépítési és karakter-animációs eszközök (megadott kulcspozíciók interpolálása, nemlineáris animáció, direkt és inverz kinematika);
- hatékony deformációs eszközök az izmok feszülése (például beszéd, mosoly) modellezésére;
- integrált részecskedinamika, valamint merev és rugalmas testek kölcsönhatásának kezelése;
- nyomásérzékeny “ecsetek” 2D-s felületre és 3D-s térben való festéshez (Maya Paint Effects, Maya Artisan);
- ruhaanyag, haj, szőrzet realisztikus modellezésének széles skálája;
- légköri, pirotechnikai, folyadékbeli effektusok kezelése;
- több rendelkezésre álló renderelő (raytracing alapú Maya szoftver renderelő, hardver és vektor renderelő, mental ray) közötti választás
- teljes programozói felhasználói felület (API) és beépített szkript szerkesztő (MEL);
- gyors adatcsere az Adobe Photoshop és Adobe Illustrator termékekkel.

318. Mi a VRML és milyen alapfunkciókat nyújt?
A virtuális valóság modellező nyelve a VRML (Virtual Reality Modeling Language), amely az 1990-es évek második felében vált szabvánnyá (ISO/IEC 14772). A VRML-lel alapvetően

térgeometriai objektumokat definiálhatunk. Ezek lehetnek például kockák, kúpok, gömbök stb.

319. Ismertesse a VRML 2 új lehetőségeit a VRML 1-hez képest!

- a virtuális térben szenzorokat helyezhetünk el, amelyekkel meghatározott eseményeket figyelhetünk és ezek bekövetkezésekor akciókat kezdeményezhetünk;
- a virtuális térben már valós fizikai viszonyokat is modellezni lehet, például gravitáció, repülés;
- a VRML fájlban hangforrásokat határozhatunk meg;
- a virtuális térben ütközést is modellezhetünk, át nem járható tárgyak is létezhetnek;
- a virtuális teret a 3D-s grafikában már ismert effektusokkal is felruházhatjuk, például speciális textúrák, köd stb.

320. Mit értünk séta (walk) alatt a virtuális térben?

A séta (walk) a virtuális valóságban azt jelenti, hogy egy program igénybevétele – melyet lejátsszó programnak nevezünk – „bejárhatjuk” a 3D-s modellteret.

321. Mit jelent a VRML fájlok lejátszása, hogyan változtathatjuk helyzetünket a virtuális térben?

A VRML fájlok lejátszásakor a virtuális térbeli helyzetünket, vagyis a nézőpont meghatározását interaktív módon, például egérrel folyamatosan vezéreljük. Ennek hatására a lejátsszó program a képernyőn folyamatosan kirajzolja a helyzetünknek megfelelő térhatású képét a modellter jelenetének.

322. Mi a POV-Ray?

A POV-Ray (Persistence of Vision Ray-Tracer) egy nyílt forráskódú, ingyenesen terjeszthető 3D-s fotorealistikus renderelő program, amely a rendereléshez a raytracing technikát alkalmazza. A program egy saját leírónyelvet használ, ezzel kell definiálni a modellter objektumait, a kamerát, a fényforrásokat és az effektusokat. A leírónyelven megfogalmazottakat POV kiterjesztésű fájlokban tárolják.

323. Melyek a POV-Ray fontosabb tulajdonságai?

- egyszerűen használható leírónyelv,
- előre definiált alakzatok, színek és textúrák, példajelenetek terjedelmes könyvtára,
- egyszerű (gömb, kocka, kúp stb.) és komplex (tórusz, poligon, Bézier-folt, forgási felület, fraktál stb.) primitív alakzatok, amelyekkel CSG-műveleteket lehet végrehajtani,
- textúrák réteg- vagy mozaikszerű egyesítése,
- több kameratípus (perspektivikus, ortogonális, halszem stb.),
- photon mapping technika a kauszikus képek kezelésére,
- Phong-féle spekuláris fényviszszaverődés kezelése,
- radiosity algoritmus lehetősége a diffúz jelenségekre,
- légköri effektusok (köd, szivárvány stb.), valamint részecske jelenségek (felhők, por, tűz, gőz) modellezése (lásd ábrákat),
- exportálás több fájlformátumba (BMP, PNG, Targa, PPM).

324. Mit nevezünk 3D-s modellezőknek, és miért fejlesztették ki azokat?

amelyekkel interaktív módon követni tudjuk a modellterben az objektumok megtervezését, a megfelelő képek megtekintését és esetleges változtatását.

325. Mire használható a Moray program?

huzalváz-modellező

A program képernyője a jelenet három 2D-s nézetét ábrázolja huzalváz formában, valamint egy 3D-s renderelt képet a kameraállásból

326. Mire alkalmas a ppModeler?

képes az elkészített jelenetet POV formátumban exportálni. Ez a modellező kiválóan alkalmas karaktermodellezésre

327. Mi a Blender?

A Blender egy ingyenes, professzionálist megközelítő 3D-s modellező és animáció-készítő program, amellyel főként filmek készítenek.

328. Ismertesse a Blender legfontosabb funkcióit!

- 3D-s objektumok modellezése poligonhálókkal, NURBS felületekkel, Bezier és B-spline görbékkel, metaball alakzatokkal, vektorgrafikus betűkészlettel,
- fejlett animációs funkciók (direkt és inverz kinematika, automatikus bőrzés, nemlineáris animációkeverés stb.),
- hang és kép szinkronizálásának támogatása,
- gyors beépített sugárkövetéses renderelő, radiosity fényterjedésszámítás,
- több felületárylító eljárás közötti választás (például Phong),
- grafikus szerkesztő interaktív tulajdonságok meghatározására,
- UV textúraszerkesztő, textúrákkal előállítható effektusok (environment mapping, reflection mapping stb.),
- procedurális textúrák,
- jelenetek réteges kezelése,
- saját tömöríthető fájlformátum (.blend), egyéb fájlformátumok támogatása (TGA, JPG, AVI, TIFF stb.).

329. Mire alkalmas a GIMP programcsomag?

A GIMP (GNU Image Manipulation Program) egy nyílt forráskódú, ingyenesen terjeszthető szoftver, amely alkalmas rajzolásra, grafikus képek előállítására és összeszerkesztésére, digitális fotók retusálására, általában profi szintű képmanipulálásra.

330. Sorolja fel a GIMP fontosabb lehetőségeit!

- estőeszközök széles választéka (ecsetek, ceruza, festékszóró stb.),
- szubpixel szintű sampling a lépcsőeffektus csökkentésére (anti-aliasing),
- több réteg használata egy képen (beleértve szerkeszthető szövegréteget is), amelyekkel algebrai kompozíciós műveleteket lehet végezni teljes alfa csatorna (Alpha Channel) támogatással az átlátszóság kezelésére,
- fejlett kijelölő eszközök (téglalap, ellipszis, fuzzy, Bézier stb.),
- transzformációs eszközök (forgatás, skálázás, nyírás, tükrözés),
- csak a rendelkezésre álló lemezterület által korlátozott képméret,
- procedurális adatbázis, amely lehetővé teszi a belső funkciók más programból való hívását,
- fejlett szkriptelési lehetőség, a legegyszerűbb művelettől a komplex képmanipulációs eljárásokig,
- több fájlformátum támogatása (a saját XCF mellett GIF, JPEG, TIFF, TGA, PCX, PDF, BMP stb.),
- nagy számú igénybe vehető plug-in, további filterek és fájlformátumok hozzáadásához